

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Systeme informatique d'aide à la mise à jour d'un répertoire homéopathique

Demotte, Béatrice

Award date:
1988

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES
NOTRE DAME DE LA PAIX
NAMUR

Système informatique d'aide
à la mise à jour d'un
répertoire homéopathique

Mémoire présenté
par Demotte Béatrice

en vue de l'obtention du grade de
licencié et maître en informatique

Promoteur : Monsieur le Professeur Jean Fichet
Année académique : 1987/1988

LBS 2724194
166750

Résumé

Le mémoire ici présenté, s'inscrit dans le cadre des activités développées par l' a.s.b.l. A.R.C.H.I.M.E.D.E., elle-même composée de chercheurs tous unis par la volonté d'étudier le plus objectivement possible, les mécanismes sous-tendant l'action thérapeutique des médecines douces et les résultats par elles obtenus.

D'ores et déjà à l'actif de l'a.s.b.l. A.R.C.H.I.M.E.D.E. : la mise au point du système informatique R.A.D.A.R. (Rapid Aid to Drug Aimed Repertorization) comportant un système navigationnel permettant au médecin homéopathe de trouver et sélectionner rapidement dans des répertoires homéopathiques donnés, les symptômes relevés sur un patient, de choisir sur base de méthodes d'aide à la décision multicritère, le(s) remède(s) approprié(s) au même patient.

Le mémoire se trouve lié au système informatique R.A.D.A.R. par la mise à jour qu'il se propose de faire sur un répertoire précisément parcouru par le système informatique R.A.D.A.R.; soit nommément, la mise à jour du répertoire de KENT, mise à jour par insertion d'éléments tirés du répertoire de BOENNINGHAUSEN.

Summary

This essay comes within the scope of the activities developed by A.R.C.H.I.M.E.D.E. asbl which is made up of researchers whose common purpose is to study, as objectively as possible, the mechanisms underlying the therapeutic action of gentle (unofficial) medicines and the results obtained.

Current achievements of A.R.C.H.I.M.E.D.E. asbl : the development and start-up of the R.A.D.A.R. computer system (Rapid Aid to Drug Aimed Repertorization) which includes a navigational system enabling homeopaths to find and select rapidly in a specific homeopathic repertory, the symptoms shown by a patient and to choose, on the basis of a multiple choice method, the appropriate remedy(ies) for the same patient.

This essay ties in with R.A.D.A.R. computer system via the up-dating it is to carry out on a repertory which is used by the R.A.D.A.R. computer system; that's to say, the up-dating of the KENT repertory by inserting data taken from the BOENNINGHAUSEN repertory.

Je tiens à remercier Monsieur le Professeur Jean Fichet de l'attention généreusement accordée.

Je tiens à remercier Monsieur Jacques Paris des nombreuses directives régulièrement prodiguées dans l'élaboration du présent travail.

Enfin, je tiens à remercier les membres du personnel de l'a.s.b.l. A.R.C.H.I.M.E.D.E.

1. INTRODUCTION

Le mémoire ici présenté, s'inscrit dans le cadre des activités développées par l'a.s.b.l. A.R.C.H.I.M.E.D.E., elle-même composée de chercheurs tous unis par la volonté d'étudier le plus objectivement possible, les mécanismes sous-tendant l'action thérapeutique des "médecines douces" et les résultats par elles obtenus.

Figure de proue des "médecines douces" les plus pratiquées aujourd'hui : l'homéopathie et l'inévitable convoitise suscitée par elle auprès des partisans de pratiques occultes.

D'où la priorité accordée à son étude et d'ores et déjà la mise au point d'un projet couvrant :

- _la construction de systèmes informatiques permettant une utilisation tant rapide qu'efficace de l'énorme système d'information sur lequel se base la démarche homéopathique
- _l'élaboration d'une banque de données cliniques destinées aux études d'efficacité et de fiabilité de l'homéopathie et au développement des connaissances en ce domaine
- _le contrôle des qualités des teintures mères employées dans la fabrication des remèdes homéopathiques.

Conçu dans le cadre du présent projet, le système informatique R.A.D.A.R. (Rapid Aid to Drug Aimed Repertorization) comporte un système navigationnel permettant au médecin de trouver et sélectionner rapidement, dans des répertoires homéopathiques donnés, les symptômes relevés sur un patient, de choisir sur base de méthodes d'aide à la décision multicritère, le(s) remède(s) approprié(s) au même patient.

Le mémoire se trouve lié au système informatique R.A.D.A.R. par la mise à jour qu'il se propose de faire sur un répertoire précisément parcouru par le système informatique R.A.D.A.R.; soit nommément, la mise à jour du répertoire de KENT(1), mise à jour par insertion d'éléments tirés du répertoire de BOENNINGHAUSEN(2).

Le paragraphe 1 rappelle quelques uns des grands principes de l'homéopathie, lesquels sont nécessaires à la compréhension du système informatif R.A.D.A.R., lui-même présenté au paragraphe 2. Le paragraphe 3 traite du pourquoi de la mise à jour ici envisagée.

1. L'homéopathie en bref

A l'origine des grands principes de l'homéopathie, un médecin allemand répondant au nom de Samuel HAHNEMANN (1755-1843).

C'est frappé par l'incohérence des explications données sur les propriétés de la quinine, dans "la matière médicale" de CULLEN, qu'il décide à l'âge de 35 ans, d'en expérimenter l'action sur lui-même. Il en prend plusieurs jours durant et bientôt ressent les symptômes d'un état fébrile intermittent identique aux fièvres précisément guéries par la quinine. Il renouvelle l'expérience, l'étend à la belladone, au mercure, constatant une réponse toujours concordante, vérifie ce faisant l'antique loi de similitude disant que " toute substance susceptible de déterminer chez l'individu sain certaines manifestations, est susceptible chez l'individu malade de faire disparaître ces mêmes manifestations " (a); soit encore SIMILIA SIMILIBUS CURENTUR.

Ses principes se résument en :

_ la loi de similitude déjà énoncée.

_ l'individualisation du malade à savoir l'obligation qu'il y a d'individualiser non seulement la maladie mais la forme qu'elle revêt chez le sujet considéré. L'individualisation du malade consiste en conséquence à définir sa constitution, à reconnaître ses réactions tempéramentales et observer ses réactions pathologiques, l'ensemble constituant la physionomie morbide du sujet.

(a) Pierre Vannier, L'homéopathie, p. 37, éd. Presses Universitaires de France, 1977

_l'individualisation du remède, traduisant la propriété que chaque remède a de produire sur l'individu sain, un ensemble de symptômes qui lui est propre, ensemble encore appelé physionomie médicamenteuse.

_l'expérimentation sur l'individu sain puisque seule à pouvoir indiquer les vertus curatives d'une substance, seule à provoquer sur l'individu sain les symptômes caractéristiques de la substance.

Le recueil des pathogénésies ou relevés par substance, de l'ensemble des réactions produites sur l'individu sain, constitue la Matière Médicale Homéopathique.

Parmi les plus célèbres aujourd'hui : l'encyclopédie en 12 volumes de T.F.A. ALLEN (3), celle en 10 volumes de C. HERING (4). La difficulté inhérente à la manipulation de telles encyclopédies est à l'origine de la constitution de répertoires symptomatiques formés de l'ensemble de tous les symptômes qu'est supposé présenté l'individu malade avec pour chacun d'eux, la liste des remèdes connus pour avoir guéri le symptôme dans des cas réels de maladie ou l'avoir produit à l'occasion d'expérimentations sur individus sains, remèdes tous affectés d'un degré d'efficacité.

Les répertoires se trouvent tous organisés de façon hiérarchisée en arbre aux noeuds ou sommets duquel correspondent les rubriques (traduisez : ensemble formé d'un intitulé et lorsqu'elle figure, une liste de remèdes attachés), tout symptôme dès lors assimilé à une hiérarchie donnée de rubriques.

En exemple : le répertoire de KENT.

Un répertoire permet d'accélérer le processus menant au choix du remède correspondant aux symptômes relevés, une matière médicale en autorise le raffinement et/ou la confirmation.

_la dynamisation de la substance médicamenteuse étant entendu que "plus une substance est diluée et soumise à des succussions (traduisez : secousses vigoureuses), plus grand est son effet thérapeutique au contraire de son effet toxique alors annulé " (a).

(a) Vikoulkas, La science de l'homéopathie, éd. du Rocher, 1984

Son application comporte essentiellement 3 temps :

- _la collecte des symptômes auprès du patient
- _la recherche dans les répertoires ad hoc, des symptômes relevés
- _l'agrégation des listes de remèdes associés aux symptômes repérés, avec pour résultat, la mise en évidence du remède dont la physionomie médicamenteuse correspondrait le plus à celle morbide établie au premier temps.

3 temps à assister autant que faire se peut par l'ordinateur et la capacité qui le caractérise à traiter d'énormes quantités de données.

Assisté, l'homéopathe pourra effectuer une navigation sélective au sein des répertoires concernés, se décharger de la construction du tableau à double entrée : symptômes et liste de remèdes associés permettant la mise en évidence du remède à la somme d'occurrences la plus élevée, envisager dès lors le traitement de listes de remèdes aussi longues soient-elles et l'utilisation de méthodes d'aide à la décision multicritère plus performantes que la simple addition d'occurrences.

2. R.A.D.A.R. plus en détails

Le système informatique R.A.D.A.R. se trouve composé d'un interface, d'un système cognitif et d'une base de connaissances, à l'image de tout système expert.

La base de connaissances est organisée en deux composants logiques reposant l'un et l'autre sur le répertoire de KENT et le répertoire SYNTHESIS(5).

Pour mémoire, le répertoire de KENT collationne par sous-système (mental, vertiges, tête, yeux...) de l'individu considéré comme un tout, l'ensemble de tous les symptômes importants qu'est censé présenter l'individu malade avec pour chacun d'eux, la liste des remèdes supposés le guérir, remèdes affectés d'un degré d'efficacité.

Le répertoire SYNTHESIS suit le même plan, ne couvrant qu'une partie des sous-systèmes alors complétés de tous les apports nouveaux qui n'auraient trouvé place dans les différents répertoires existant antérieurement.

Les deux répertoires couvrent à eux seuls 1574 remèdes, 60000 symptômes.

Premier composant : la base de connaissances passives (ou fichier des faits), formée de l'ensemble des symptômes issus des deux répertoires, organisée de façon hiérarchisée en arbre conformément à la structure des deux répertoires.

Deuxième composant : la base de connaissances actives (ou règles de production), à ce jour uniquement basée sur les deux répertoires mentionnés, d'où la simplicité qui la caractérise, un répertoire ne donnant par symptôme que la liste des remèdes censés le guérir, avec leurs degrés d'efficacité respectifs.

Témoin les deux règles ici énoncées, avec S mis pour symptôme :

Si S alors visualiser les remèdes susceptibles de guérir S,
avec leurs degrés d'efficacité respectifs

Si S alors garder S ainsi que la liste des remèdes
susceptibles de guérir S, avec leurs degrés
respectifs.

Le système cognitif contient le mécanisme de raisonnement permettant à la fois de répondre aux questions et résoudre les problèmes posés par l'utilisateur.

L'information arrivant au système cognitif peut se résumer en un tableau à double entrée : symptômes observés et remèdes censés faire disparaître ces mêmes symptômes, de sorte que le mécanisme de raisonnement conduisant au remède susceptible de guérir le patient relève de la méthodologie d'aide à la décision multicritère. R.A.D.A.R. propose deux méthodes habituellement prônées par les homéopathes :

_celle classant les remèdes conformément à leurs nombres d'occurrences

_celle classant les remèdes selon leurs sommes de degré d'efficacité.

L'interface permet

_l'introduction par le médecin-utilisateur, des symptômes observés chez le patient;

soit encore l'acquisition des données.

_l'enregistrement des symptômes, la modification d'une liste de symptômes enregistrés, l'appel à une méthode d'aide à la décision, etc...

soit encore la communication homme-machine

3. Le pourquoi de la mise à jour envisagée

L'intérêt de la mise à jour envisagée réside dans le fait que le répertoire dressé par VON BOENNINGHAUSEN (1785-1864) apparaît comme complémentaire du répertoire de KENT dans le traitement de cas difficiles;

témoin l'un deux emprunté au docteur Jean-Claude Grégoire qui s'en explique :

" il s'agissait d'un enfant de trois ou quatre ans qui présentait, notamment sous une oreille, une fissure très marquée. Le symptôme ne figurant pas au répertoire de KENT où l'on ne trouve p.289 que **Eruptions, below ears, boils ; Calc.**, je décidai de m'assister du BOENNINGHAUSEN.

Je choisis les rubriques **beneath ears**
skin, cracks

Je décidai de ne retenir que les remèdes au moins au plus haut degré dans l'une des deux rubriques, soit pour

beneath ears ; BAR-C., BELL.

skin, cracks ; CALC., GRAPH., PULS., SEP., SULPH.

En additionnant les points des sept remèdes, je trouvais pour bar-c. 8, bell. 5, calc. 9, graph. 5, puls. 8, sep. 9, sulph. 9. Calcarea, sepia et sulph ressortaient nettement avec Baryta-carb. et Pulsatilla, ce qui me composait pour ce sytôme local remarquable, une rubrique de cinq remèdes que je comparai au reste du tableau symptomatique, à l'aide cette fois du répertoire de KENT.

Calcarea fut alors prescrit, avec succès ! "

Le répertoire de BOENNINGHAUSEN collationne par sous-système (mental, siège des symptômes, sensations et plaintes, sommeil et rêves, circulation et fièvres, modalités d'aggravation et d'amélioration), les symptômes relevés sur l'individu malade ainsi que pour chacun d'eux, la liste des remèdes supposés le guérir, remèdes flanqués d'un degré d'efficacité.

Ajouté aux six sous-systèmes : un chapitre traitant de la concordance des remèdes, résultat d'une comparaison des effets de tous les remèdes.

A l'origine du répertoire dressé par BOENNINGHAUSEN : la méthode qu'il a lui-même développée.

A ses dires, un symptôme pour être complet se doit d'être caractérisé par (a) une localisation, (b) un type de sensation, (c) des modalités d'aggravation, d'amélioration tant horaires que de lieux et de circonstances.

Compte tenu de la rareté avec laquelle un symptôme figure à l'état complet dans une matière médicale, dans un recueil d'observations de malades, BOENNINGHAUSEN prône la reconstitution des parties manquantes, le raisonnement par analogie encore appelé généralisation des modalités : premier point caractéristique de sa méthode.

Il établit fort heureusement que les modalités d'aggravation et d'amélioration ne se limitent pas à un symptôme particulier mais s'appliquent à l'entièreté du cas étudié.

Par ailleurs, BOENNINGHAUSEN souligne l'importance des symptômes dits concomitants ou accessoires par l'individualisation qu'ils procurent à la physionomie tant médicamenteuse que morbide : deuxième point caractéristique de sa méthode.

Concrètement l'homéopathe obtient à l'examen d'un cas, des parties d'un très petit nombre de symptômes complets, parties qu'il doit alors rassembler voire compléter.

Suit la recherche des symptômes dans le répertoire de BOENNINGHAUSEN et l'agrégation des listes de remèdes attachés aux symptômes, agrégation identique à celle pratiquée par HAHNEMANN. Soit finalement une différence au niveau de la première étape, consistant pour HAHNEMANN à collecter les symptômes, pour BOENNINGHAUSEN à collecter les symptômes puis rassembler voire compléter.

2. SPECIFICATIONS

Est demandée la mise à jour du répertoire de KENT par _insertion de groupes de rubriques, traduisez : insertion groupée de rubriques à la fois adjacentes et de même niveau au sein du répertoire de BOENNINGHAUSEN, en un endroit donné du répertoire de KENT.

Le groupe est inséré avec ou sans l'entièreté des rubriques dérivées des rubriques le constituant;

_fusion de paires de rubriques, traduisez : insertion des remèdes attachés à une rubrique du répertoire de BOENNINGHAUSEN, parmi les remèdes attachés à une rubrique du répertoire de KENT, étant entendu que :

l'accès aux répertoires qu'il soit de BOENNINGHAUSEN pour la délimitation d'un groupe, la localisation d'une rubrique, de KENT pour la localisation de l'insertion, s'effectue de façon hiérarchisée conformément à la structure des deux répertoires (on parle encore d'accès direct),

la localisation de l'insertion ne subit aucune restriction,

l'introduction des données relatives aux éléments à insérer ou fusionner s'effectue en une voire plusieurs sessions au gré de l'utilisateur.

A cet effet deux chapitres du répertoire de BOENNINGHAUSEN ,les chapitres VERTIGO et HEAD EXTERNAL, seront encodés selon des conventions à établir.

L'implémentation s'effectuera en C sur un ordinateur de la marque Olivetti AT&T 3B2/400.

3. DEMARCHE

S'impose à priori l'étude comparative des deux répertoires de KENT et de BOENNINGHAUSEN, partant la définition d'une conversion rendant les rubriques (ou parties de rubriques) issues du répertoire de BOENNINGHAUSEN conformes aux rubriques (ou parties de rubriques) du répertoire de KENT.

Suit l'étude des deux versions encodées du répertoire de KENT :

_la première, confondue aux conventions d'écriture près, à la version originale du répertoire de KENT; y correspond la version encodée du répertoire de BOENNINGHAUSEN, aux conventions d'écriture à établir dans le cadre du mémoire,

_la deuxième, éclatée en différents fichiers ayant chacun trait à une information propre, un éclatement voulu pour la performance de R.A.D.A.R. et réalisé au départ de la première version; s'en inspirent les moyens mis en oeuvre pour la performance de l'accès hiérarchisé au répertoire de BOENNINGHAUSEN encodé, accès nécessaire à la mise à jour à concevoir.

En dérive la conception de la mise à jour des deux versions encodées du répertoire de KENT, concrètement la mise à jour de la première version encodée, elle-même alors éclatée en une deuxième version encodée, par construction mise à jour.

3.1 ETUDE DE LA STRUCTURE DU REPERTOIRE DE BOENNINGHAUSEN

Le répertoire de BOENNINGHAUSEN est organisé de façon hiérarchisée en arbre aux noeuds et sommets duquel correspondent des rubriques seules ou en cascade, sachant une rubrique

_formée d'un intitulé éventuellement suivi d'un Bloc-remèdes et/ou d'un Bloc-renvois,

_de niveau signifié par l'impression ou l'accolement à une marge particulière de l'intitulé.

3.1.1 Niveaux et intitulés de rubriques

Les rubriques de niveau 1 sont celles d'intitulé entièrement écrit en majuscules grasses;

soit encore et plus précisément, en annexe 1, les intitulés à la gauche desquels figure un nombre.

En exemple :

LARYNX AND TRACHEA.

Les rubriques de niveau 2 sont celles d'intitulé entièrement écrit en majuscules fines;

soit encore et plus précisément, en annexe 1, les intitulés à la droite desquels figure un nombre, exception faite des intitulés déjà considérés de niveau 1.

En exemple :

PATHOLOGICAL TYPES

Les rubriques de niveau 3 et plus présentent la particularité de parfois se succéder sur une même ligne, de former en quelque sorte une cascade de rubriques, les premières se réduisant à un intitulé, la dernière figurant en outre un Bloc-remèdes et/ou un Bloc-renvois.

En toute généralité, les rubriques de niveau 3 sont celles d'intitulé accolé à la première marge, un intitulé formé d'un premier mot toujours gras et de ce qui le suit directement jusqu'aux deux points débutant le Bloc-remèdes ou jusque

l'ensemble formé d'une virgule directement suivie du mot *see* débutant le Bloc-renvois pour autant qu'aucun changement d'écriture n'ait été observé auquel cas l'intitulé (et par la même la rubrique de niveau 3) s'arrête au changement d'écriture.

Qu'entendre par changement d'écriture ?

_la rencontre d'une parenthèse ouvrante

_la rencontre du premier caractère italique pour autant qu'il existe et ne soit pas confondu avec le premier caractère de l'intitulé considéré.

Un changement d'écriture est à considérer comme précision de l'intitulé au sein duquel il figure, à l'image d'une rubrique de niveau $i-1$ précision d'une rubrique de niveau i .

En exemples :

Extravasation of blood: *Arg., fer.,*

y correspondent la rubrique de niveau 3, d'intitulé Extravasation
la rubrique de niveau 4, d'intitulé of blood

Cutting. (L): *Arg., bu/, kali-n., Mane.,
nit-ac., thu.*

y correspondent la rubrique de niveau 3, d'intitulé Cutting
la rubrique de niveau 4 d'intitulé L

Noter ici qu'au sein même du répertoire et ainsi définies, les rubriques de niveau 1 sont parfois directement suivies de rubriques de niveau 3.

En exemple :

INGUINAL AND PUBIC REGION

Inguinal Region: *Aco., am-m., aur., bap.,
clem., cocl., colo., gray., gwei., kali-c.,
lob., lyc., nux-v., sul-ac., tab., ver-a.*

Soit une rupture de niveaux !

Pour y pallier, des rubriques de niveau 2, fictives et chacune réduite à l'intitulé "Miscellaneous" seront intercalées lorsque besoin en sera.

D'où la version corrigée de l'annexe 1, désormais seule considérée et donnée en annexe 2.

Les rubriques de niveau i (avec $i \geq 4$) sont celles d'intitulé accolé à la $(i-2)$ ème marge ou encore celles commençant là où s'arrête un intitulé (et par la même, une rubrique de niveau $i-1$) lorsque l'arrêt n'a été signifié ni par les deux points débutant le Bloc-remèdes ni par l'ensemble formé d'une virgule directement suivie du mot *see* débutant le Bloc-renvois.

L'intitulé d'une rubrique de niveau i est alors formé d'un premier mot (accolé à la $(i-2)$ ème marge dans le premier cas, suivant direct de l'intitulé rubrique de niveau $i-1$ dans le deuxième cas) et de ce qui le suit directement jusqu'aux deux points débutant le Bloc-remèdes ou jusque l'ensemble formé d'une virgule directement suivie du mot *see* débutant le Bloc-renvois pour autant qu'aucun changement d'écriture n'ait été observé auquel cas l'intitulé (et par la même la rubrique de niveau i) s'arrête au changement d'écriture.

Veillez à ne pas confondre la définition d'une abréviation avec un intitulé formé de la même abréviation, une confusion née du fait que la définition d'une abréviation, en d'autres termes, le terme normalement écrit et sa correspondance abrégée se présente

_ou sous forme littérale, à la suite immédiate d'une rubrique de niveau 2

_ou sous forme camouflée en ce sens que rien ne la différencie d'intitulé de rubrique, rien si ce n'est qu'il s'agit de la première apparition de l'abréviation dans l'ensemble des rubriques de niveau 2 et plus, directement descendantes de la rubrique de niveau 1, elle-même ascendante de la rubrique englobant l'abréviation.

En exemple :

FACE

B. stands for Bones.

One side, unilateral: Aco., am-c., am-m.,
arg-n., csus., cham., colch., COLO.,
CON., gel., kre., mez., Nat-m.,
NUX-V., Pho., Plat., SPI., stap.,
ver-a.

Right, R.: Aco., agr., Ag-c., alu., am-c.,
Ans-m., anac., ant-c., ant-f., ap., arg.,
ara., Ara., asal., aset., Awe., bor-a.

Sa syntaxe :

<abbreviation>::=

<majuscule> %<majuscule>% . \$

<majuscule> %<minuscule>% .

<majuscule>::=

A\$B\$C\$D\$E\$F\$G\$H\$I\$J\$K\$L\$M\$N\$O\$P\$Q\$R\$S\$T\$U\$V\$W\$X\$Y\$Z

<minuscule>::=

a\$b\$c\$d\$e\$f\$g\$h\$i\$j\$k\$l\$m\$n\$o\$p\$q\$r\$s\$t\$u\$v\$w\$x\$y\$z

Exemple récapitulatif avec marges et niveaux figurés :

- ³ Extending limbs: Ara.
³ Face, heat of with: Hep., merc., nux-v.
⁴ pale: Can., pho.
⁴ red, shiners on: Colo.
³ Fasting, hunger, etc.: Aco., asar., Calc-c.,
 croc., dul., mar., nux-v., plat., sep.,
 stap.
³ Fat foods: Carb-v., chin., colch., nat-m.,
 Pul.
³ Fever, before: Chin., spo.
³ Flatulent foods: Bry., calc-c., Chin.,
 kali-c., Lyc., Petr., pul., sep., ver-a.
³ Flatus, discharge of, before: Chin., con.,
 Grap., Lyc., plat., Pul., ran-b., rhe.,
 sil.
⁴ during: Con., ox-ac., plat., Pul.
⁴ after: Cam., grap., sul-ac.
³ Frozen foods: Calc-p., ip., pul.
³ Gout, with: Daph.
³ Hands, yellow: Sil.
³ Headache with: Hyo., pho.
³ Heat of body, with: Ara., carb-v., cocl.,
 colo., grat., Nat-c., nat-m.
³ Hemorrhoids from: Carb-v., cham., colo.,
 lach., NUX-V., PUL., soc-a., Sul.
³ Herring, after: Nat-m.
³ Humor out of, as of: Cic., kre., sul.
³ Hypochondriasis, with: Caloc-a., cham.,
 NUX-V., stan., stap., sul.
³ Hysteria, with: Ara., bell. bry., Cocl.,
 IGN., Ip., mag-m., mos., Nux-v., stan.,
 stra., sul., cal., ver-a.
³ Indignation: Ant-c., bell. bry., carb-v.,
 colo., hep., Nux-v., PUL., sul.
- ³ Injuries to: Arn., bry., carb-v., lach.,
 rhus-t.
³ Inspiration: Am-m., bry., kali-bi., lyc.,
 mag-m., plat., rhus-t., sele., tab.
³ Laughing: Ara., colo., con., lyc., nux-v.,
 plb.
³ Leaning on a sharp edge: Ran-b., samb.
³ Lead poisoning: Alu., arg-n., bell., cham.,
 Op., Plat.
³ Leucorrhoea, with: Caus., con., grap.,
 ign., kre., Lyc., mag-c., MAG-M., nat-c.,
 nat-m., PUL., Sil., sul., zin.
⁴ during: Grap., sep.
³ Lie down, impulse to, with: Ant-t., nux-v.
³ Limbs, lame with: Carb-v.
⁴ pains with: Colo., sec-c.
³ Lumbar pain, with: Alu., am-c., bar-c.,
 caloc-a., caus., cham., kali-c., kre., mag-m.,
 nat-m., Nux-v., pho., Pul., sec-c.,
 Sul.
³ Lying: Amb., amy-n., ars., asaf., bell., can.,
 carb-a., caus., cham., colo., con., dig.,
 euphor., fer., glo., mag-c., pho., pho-ac.,
 plb., pru., pul., rho., rhus-t., sep., spi.,
 stan., thu., zin.
⁴ back, on: Lyc., pod.
⁴ bed, in: Amb., Cham., Chin., cocl.,
 cof., dro., ign., mar., mos., nux-m.,
 nux-v., Pul., Rhus-t., sabi., spi., stap.,
 val., ver-a.
⁴ side, on: Alo., bry., ign.
⁵ left: Carb-v., glo., nat-c., par.,
 pho., pul., rhus-t.
⁵ right: Caus., merc., spo.
⁵ painful: Nux-v., spo.
⁵ painless: Bry., cham., ign.

3.1.2 Bloc-remèdes

Par Bloc-remèdes, on entend tout ensemble du répertoire de BOENNINGHAUSEN formé de deux points directement suivis d'une liste de remèdes.

Les remèdes y sont :

_rangés alphabétiquement

_séparés par des virgules

_de degré signifié par l'écriture particulière des lettres le formant, ainsi :

les lettres capitales pour les remèdes de degré 5,

les lettres grasses pour les remèdes de degré 4,

les lettres italiques pour les remèdes de degré 3,

les lettres romaines pour les remèdes de degré 2,

les lettres romaines entre parenthèses pour les remèdes de degré 1

_de syntaxe conforme à :

<remède>::=

<car> %<car>% \$

<car> %<car>% . \$

<car> %<car>% - <car> %<car>% . \$

<car> %<car>% - <car> %<car>% - <car> %<car>% .

<car>::=

<majuscule> \$

<minuscule>

En exemple :

Squeezing, compression: Bell, calad.

calc-c., carb-v., (caus.), chel., cina,

colch., dig., dro., grap., kali-bi., kali-c.,

kali-n., mez., mos., nat-c., petr., pho-ec.,

pul., rho., sil., spi., stro., sul.

U.: Aco., alu., am-m., Anac., aur.,

caus., croz., kali-n., lau., lyc. (r),

mar. (l), mar-ec., pho-ec., sabi., spi.,

spo., sul., thu.

3.1.3 Bloc-renvois

Par Bloc-renvois, on entend tout ensemble du répertoire de BOENNINGHAUSEN formé de ", see" ou "compare" directement suivis d'une liste d'intitulés de rubriques référées.

sauf exception, les intitulés référés s'écrivent et se délimitent comme lorsque non référés si ce n'est qu'ils ne s'accolent à aucune marge particulière, que les intitulés de niveau 1, 2 et 3

sont tous écrits en caractères gras.

D'où la nécessité d'un complément aux règles de détermination du niveau d'un intitulé, un complément seulement valable au sein du Bloc-renvois.

Soit explicitement la prise en compte de trois cas de figure :

a. La rubrique de niveau 2, ascendante de la rubrique considérée a un intitulé de l'une des deux formes "X-Cross Reference", "X and Y-Cross Reference" (entendant par X et Y, des intitulés de rubriques) auquel cas vérifier si l'intitulé référé correspond à l'intitulé d'une rubrique descendante de la rubrique X(Y) lorsque X(Y) est de niveau ≥ 2 , de la rubrique d'intitulé "Miscellaneous" descendante de la rubrique X(Y) lorsque X(Y) est de niveau 1.

En exemple :

ORBITS

Orbits, in general: Alu., am-m., anac., Boring: Calc-c., hep.

ang., ant-c., ara., ara., asaf., aur., bar-c..

Bell., biem., bov., bry., calc-c., cam..

chel., chin., coel., colo., con., cup., dig..

hell., hep., hys., ign., iod., lach., lau..

led., lyc., mag-c., men., merc., mez..

mur-ac., nit-ac., nux-m., Nux-v., old..

par., pha., pho-ac., Plat., plb., pul..

rho., rhus-t., rut., sars., sele., seng., sep..

SPL. spo., Stan., stro., sul., sul-

VAL., ver-a., verò.

Bruised, beaten, pain as if: Nux-v.,
rhus-t.

Burned, as if: Bry.

Burning: Men., sul.

ORBITS—CROSS REFERENCE

Beaten, see **Bruised.**

Squeezing, see **Pressing.**

Raw, see **Excoriative.**

b. La rubrique de niveau 2, ascendante de la rubrique considérée a comme intitulé "Cross Reference" auquel cas vérifier si l'intitulé référé correspond à l'intitulé d'une rubrique descendante de la rubrique d'intitulé "Miscellaneous", ayant même rubrique ascendante que la rubrique d'intitulé "Cross Reference" considérée.

En exemple :

PERINEUM

Perineum: AG-C., ALU., am-m., Ant-c.,
Ant-t., ars., asa/, bell., bov., bry.,
calc-c., can., CARB-A., CARB-V.,
Caus., chin., CYC., grap., hep., ign.,
Lyc., mag-m., merc., mez., mur-ac.,
Nux-v., paeon., Petr., pho., Plb., Rho.,
rhus-t., seng., sep., spi., SUL., Tarx.,
tha.

Abcess: Ant-c., ant-t., Caus., cro-t., paeon.

Biting as if: Ag-c.

Blood-boils: Ant-c.

Boring, Merc., spi.

Bruised feeling: Alu., spi.

Burning: Alu., ant-c., ant-t., cyc.,
mur-ac., nit-ac., nux-v., plb., rho., sil.

Burrowing: Carb-a., carb-v.

Contraction, cramp, clamping, etc.: Canth.,
lyc., Sul., thu.

Corrosive, gnawing sensation: Ag-c.

CROSS REFERENCE

Clamping pain, see Compressive.

Eruption, see Pimples.

Cramp, see Contraction.

Gnawing sensation, see Co

c. La rubrique de niveau 2, ascendante de la rubrique considérée a un intitulé à la fois différent de "X-Cross Reference", "X and Y-Cross Reference" et "Cross Reference" auquel cas vérifier si l'intitulé correspond à l'intitulé d'une rubrique descendante de la rubrique de niveau 2, elle-même ascendante de la rubrique considérée.

En exemple :

AGGRAVATION

Air, cold: Agar., bell., nit-ac., rhus-t.

Driving: Ars-i.

inhaling, thro nose: Mar.

Eating, before: Sil.

open: Agar., am-m., carb-a., Con.,
euphor., lyc., mang., spi.

while: Con., grap., mag-c., sul-ac.,
verb., zin.

Awaking: Bell., bor., carb-a., kali-bi.,
nux-e., pul., rho.

Compare Chewing.

Biting, see Teeth.

after: Carb-a., canth., cin., con., mag-c.,
op., plb., sil., zin.

Boring in ear with finger: Bor., chel., rut.

Compare Swallowing.

Chewing: Alu., am-c., arg., can., Grap.,
hep., nat-m., nit-ac., nux-m., nux-v.,
seng.

Eructation: Bell., sul.

Chill, see Fever, Chill, Concomitants.

Exanthemata, after: Bell., bry., carb-v.,
Colch., hep., lyc., Men., merc., PUL.,
Rhus-t., sul.

Dans les trois cas :

_s'il y a correspondance, le niveau de l'intitulé référé est celui de l'intitulé correspondant

_s'il n'y a pas correspondance, il s'agit nécessairement ou d'un intitulé de niveau 1, 2 ou d'une exception aux règles établies.

Une succession d'intitulés référés de niveau respectivement égal à i , $i+1$, $i+2$... constitue un renvoi, une cascade en d'autres termes d'intitules référés.

Un renvoi prend fin au point signifiant la fin du Bloc-renvois lorsqu'aucune rencontre préalable d'un intitulé de niveau inférieur ou égal au dernier intitulé considéré et différent de "Aggravation", "Amelioration", "Conditions", "Cross Reference", "Concomitants" (en forme abrégée et/ou combinée).

En exemples :

a_Company wants, see ³Company, ⁴desires.
soit un renvoi.

b_Absorbed, see ³Introspective, ³Thought.
soit deux renvois.

c_Consolation, agg., see ³Sympathy, ¹⁽²⁾agg.
soit un renvoi.

d_Indignation, see ³Emotions, ¹⁽²⁾Conditions in general.
soit un renvoi.

e_Conversation, see ³Talking under ¹⁽²⁾conditions in general.
soit un renvoi.

f_Beating, see ³Throbbing and ³Pulsation
soit deux renvois.

Noter :

_l'inversion occasionnelle (témoin, l'exemple d)

_l'abréviation occasionnelle d'intitulés de niveau 1 et 2

(témoins les exemples c, d et e où agg. se substitue à conditions of aggravation and amelioration in general)

_le regroupement occasionnel sous un même intitulé, via under, and, on de deux intitulés (témoins, les exemples e et f).

(2) mis pour le niveau de la rubrique d'intitulé Miscellaneous

3.2. ETUDE DE LA STRUCTURE DU REPERTOIRE DE KENT

Le répertoire de KENT est organisé de façon hiérarchisée en arbre aux noeuds et sommets duquel correspondent des rubriques seules, sachant une rubrique

_formée d'un intitulé éventuellement suivi d'un Bloc-remèdes et/ou d'un Bloc-renvois

_de niveau signifié par l'impression ou l'accolement à une marge particulière de l'intitulé.

3.2.1 Niveaux et intitulés de rubriques

Les rubriques de niveau 1 sont celles d'intitulé assimilé à un intitulé de chapitre, des rubriques en conséquence toutes réduites à l'intitulé même.

Les rubriques de niveau i (avec $i \geq 2$) sont celles d'intitulé accolé à la $(i-2)$ ème marge, un intitulé prenant fin aux deux points débutant le Bloc-remèdes ou à l'ensemble formé d'une parenthèse ouvrante directement suivie du mot *see*, débutant le Bloc-renvois.

3.2.2 Bloc-remèdes

Par Bloc-remèdes, on entend tout ensemble du répertoire de KENT formé de deux points directement suivis d'une liste de remède.

Les remèdes y sont :

_rangés alphabétiquement

_séparés par des virgules

_de degré signifié par l'écriture particulière des lettres le formant; ainsi :

les lettres capitales pour les remèdes de degré 3

les lettres italiques pour les remèdes de degré 2

les lettres romaines pour les remèdes de degré 1

_de syntaxe conforme à celle définie en page 15.

3.2.3 Bloc-renvois

Par Bloc-renvois, on entend tout ensemble du répertoire de KENT formé de " (see" directement suivi d'une liste d'intitulés de rubriques référées , elle-même directement suivie de ")".

3.3 CONVERSION

Les rubriques tirées du répertoire de BOENNINGHAUSEN auront à subir une conversion portant sur trois points :

a. la scission des cascades, le répertoire de KENT étant organisé de façon hiérarchisée en arbre aux noeuds et sommets duquel correspondent des rubriques seules par opposition au répertoire de BOENNINGHAUSEN, organisé de façon hiérarchisée en arbre aux noeuds et sommets duquel correspondent des rubriques seules ou en cascade.

Possibilité sera donc donnée à l'utilisateur de scinder une cascade, en d'autres termes, scinder un intitulé chaque fois que figure un changement d'écriture;

soit pour l'intitulé ici souligné, une scission possible en sensation.

Swimming sensation: Ham., jach.

temples, in: Colo., cep., cyc.

Possibilité sera encore donnée de revoir les niveaux des rubriques dérivées de cascade scindée pour autant que le niveau proposé ne soit pas inférieur au niveau d'origine ni supérieur de plus d'une unité au dernier niveau enregistré;

soit pour la rubrique d'intitulé temples, in dérivée de la rubrique d'intitulé sensation : 4 et 5 pour valeurs de niveau possible.

En règles générales :

possibilité sera donnée de scinder une cascade,

possibilité sera donnée de revoir un niveau de rubrique lorsque se sera produit au niveau directement inférieur soit une scission, soit une modification de niveaux.

b. la substitution des abréviations

Sera demandée la correspondance de toute abréviation, les correspondances nulles étant acceptées au même titre que d'autres pour résoudre le problème posé par les définitions camouflées d'abréviation.

c. la conversion des degrés attachés aux remèdes, KENT envisageant trois valeurs au contraire des cinq envisagées par BOENNINGHAUSEN.

La table de conversion ici utilisée est proposée par le docteur M. L. Dhawale (a).

BOENNINGHAUSEN	KENT
5	3
4	2
3	2
2	1
1	1

(a) Ramanlal P. Patel, The Art of Case Taking and Practical Repertorisation in Homeopathy, p.68, éd. Sai Homeopathic Book Corporation, India

3.4 PREMIERE VERSION ENCODEE DU REPERTOIRE DE KENT : ETUDE ET CORRESPONDANCE DANS BOENNINGHAUSEN

3.4.1 Etude

La première version encodée du répertoire de KENT reprend l'entièreté des rubriques y figurant

_dans l'ordre de leur apparition étant entendu que chaque rubrique se doit d'être directement suivie de l'entièreté des rubriques qui lui sont dérivées. En d'autres termes, sachant les rubriques de niveau 2 et plus réparties par page du répertoire de KENT sur deux colonnes, toutes les rubriques de la colonne gauche dérivées d'une même rubrique de niveau 1 précéderont directement toutes les rubriques de la colonne droite dérivées de la même rubrique de niveau 1.

Entre les rubriques des colonnes gauche et droite figure un marqueur de début de colonne, de forme :

numero de page l pour les débuts de colonne gauche

numéro de page r pour les débuts de colonne droite

_sous une forme fonction des composants de la rubrique, soit :
pour une rubrique formée d'un intitulé seul, la forme :

& niv intitulé : -CRT

pour une rubrique formée d'un intitulé et d'un Bloc-remèdes, la forme :

& niv intitulé : -CRT * cod deg % cod deg % : -CRT

pour une rubrique formée d'un intitulé et d'un Bloc-renvois, la forme :

+ niv intitulé Bloc-renvois : -CRT

pour une rubrique formée d'un intitulé et d'un Bloc-renvois et d'un Bloc-remèdes, la forme :

+ niv intitulé Bloc-renvois : -CRT * cod deg % cod deg % : -CRT

où

_niv a pour valeur, le niveau attribué à la rubrique d'intitulé considéré dans le répertoire de KENT

_cod a pour valeur, deux caractères identifiant le remède considéré

_deg a pour valeur, le degré attribué au remède considéré dans le répertoire de KENT

3.4.2 Correspondance dans BOENNINGHAUSEN

La version encodée du répertoire de BOENNINGHAUSEN reprend l'entièreté des rubriques y figurant

_dans l'ordre de leur apparition étant entendu que chaque rubrique se doit d'être directement suivie de l'entièreté des rubriques lui étant dérivées. En d'autres termes, sachant les rubriques de niveau 3 et plus réparties par page du répertoire de BOENNINGHAUSEN sur deux colonnes, toutes les rubriques de la colonne gauche dérivées d'une même rubrique de niveau 2 précéderont directement toutes les rubriques de la colonne droite dérivées de la même rubrique de niveau 2.

_sous une forme fonction des composants de la rubrique, soit :
pour une rubrique formée d'un intitulé seul, la forme :

& niv intitulé : -CRT

pour une rubrique formée d'un intitulé et d'un Bloc-remèdes , la forme :

& niv intitulé : -CRT * rem % , rem % : -CRT

pour une rubrique formée d'un intitulé et d'un Bloc-renvois, la forme :

+ niv intitulé Bloc-renvois : -CRT

pour une rubrique formée d'un intitulé et d'un Bloc-remèdes et d'un Bloc-renvois, la forme :

+ niv intitulé Bloc-renvois : -CRT * rem % , rem % : -CRT

où

_niv a pour valeur, le niveau attribué à la rubrique d'intitulé considéré dans le répertoire de BOENNINGHAUSEN

_rem a pour valeur, le remède considéré tel qu'il apparait dans la version originale du répertoire de BOENNINGHAUSEN

_le premier caractère italique de tout intitulé se voit précéder de [

_tout remède formé de caractères italiques se voit précéder de [

_tout remède formé de caractères gras se voit précéder de !

3.5 DEUXIEME VERSION ENCODEE DU REPERTOIRE DE KENT : ETUDE ET CORRESPONDANCE DANS BOENNINGHAUSEN

3.5.1 Etude

A l'origine de la deuxième version encodée du répertoire de KENT : l'impérative performance de R.A.D.A.R., appelé tant à parcourir le répertoire de KENT qu'à effectuer le travail mathématique de répertorisation, deux fonctions dotées chacune d'options très diverses.

La deuxième version encodée du répertoire de KENT s'établit au départ de la première version encodée. Elle correspond à un éclatement de la première version encodée en 8 fichiers ayant chacun trait à une information propre.

Soit :

_le fichier des rubriques, copie de la première version encodée, sans Bloc-remèdes ni symbole les délimitant

_le fichier listes des remèdes reprenant dans l'ordre de leur apparition dans la première version encodée, les remèdes attachés aux intitulés, remèdes ici codés numériquement

_le fichier degrés des remèdes reprenant dans l'ordre de leur apparition dans la première version encodée, les degrés attachés aux remèdes

_le fichier table des pages reprenant par marqueur de début de colonne, dans l'ordre de leur apparition dans la première version encodée, l'adresse du marqueur dans le fichier des rubriques, le niveau KENT et l'adresse dans les fichiers correspondants de la rubrique immédiatement suivante

_le fichier dictionnaire de niveau 1 reprenant le dictionnaire de niveau 1

_le fichier dictionnaires de niveau 2 reprenant l'ensemble des dictionnaires de niveau 2

_le fichier dictionnaires de niveau 3 reprenant l'ensemble des dictionnaires de niveau 3

_le fichier dictionnaires de niveau supérieur à 3 reprenant l'ensemble des dictionnaires de niveau supérieur à 3

sachant :

_le dictionnaire de niveau 1 formé des caractéristiques de toutes

les rubriques de niveau 1

_un dictionnaire de niveau i ($i > 1$) formé des caractéristiques de toutes les rubriques de niveau i , descendantes d'une même rubrique de niveau $i-1$

entendant par caractéristiques d'une rubrique :

_de niveau 1, 2, 3 et plus

a. l'adresse dans le fichier des rubriques, de l'intitulé attaché

b. l'adresse dans le fichier listes des remèdes, de la liste des remèdes attachés

c. la longueur de la liste des remèdes attachés

d. l'adresse dans le fichier dictionnaires de niveau correspondant, du dictionnaire formé des caractéristiques de toutes les rubriques directement dérivées de la rubrique considérée (on parle encore de sous dictionnaire attaché)

e. le nombre de rubriques du sous dictionnaire attaché

_de niveau 1, 2, 3

f. la longueur du sous dictionnaire attaché

g. son numéro d'ordre textuel parmi les rubriques de niveau 1 lorsque son niveau vaut 1, parmi les rubriques de niveau i dérivées d'une même rubrique de niveau $i-1$ lorsque son niveau i est supérieur à 1

h. le pointeur vers la clé lui correspondant, sachant une clé par définition formée de la racine de l'intitulé strictement nécessaire à le différencier d'autres intitulés.

Les dictionnaires de niveau 1, 2 et 3 ont pour organisation interne, le résultat d'une juxtaposition de deux tables :

_l'une (table 1) reprenant par rubrique concernée, dans l'ordre de leur apparition au sein du répertoire de KENT (ordre textuel), les caractéristiques a. b. c. d. e. f.

_l'autre (table 2) reprenant par rubrique concernée, dans l'ordre alphabétique des intitulés, les caractéristiques g. h.

Les deux tables ainsi juxtaposées se trouvent suivies d'une zone dite variable parce que formée de l'ensemble des clés correspondant aux intitulés des rubriques concernées.

On parle encore de "partie rubrique" pour l'ensemble formé des deux tables, de "partie clé" pour la zone dite variable.

Soit pour un dictionnaire de niveau 1, 2 et 3, la configuration interne suivante :

table 1	table 2
zone variable	

L'ordre des dictionnaires au sein du fichier dictionnaire(s) de niveau correspondant, respecte l'ordre d'apparition des rubriques y figurant dans le répertoire de KENT.

Les dictionnaires de niveau supérieur à 3 se groupent par ensemble formé des caractéristiques de toutes les rubriques de niveau 4 et plus descendantes d'une même rubrique de niveau 3. L'organisation interne à l'ensemble ainsi défini, vérifie la règle disant qu'au plus tôt un dictionnaire de l'ensemble est clôture dans le répertoire de KENT, au plus tôt ce dictionnaire et par là, les caractéristiques a. b. c. d. e. des rubriques y figurant sont reportées dans le fichier dictionnaires de niveau supérieur à 3. L'ordre des ensembles reportés dans le fichier dictionnaires de niveau supérieur à 3 respecte l'ordre d'apparition des rubriques y figurant dans le répertoire de KENT.

3.5.2 Correspondance dans BOENNINGHAUSEN

L'objectif étant de contribuer à la seule performance de l'accès direct ou hiérarchisé, lui-même nécessaire à la mise à jour à concevoir, la constitution à part entière d'une deuxième version encodée du répertoire de BOENNINGHAUSEN à l'image de ce qui avait été fait pour le répertoire de KENT, paraissait exagérée. Seule s'avérait indispensable la constitution de fichiers dictionnaires qui combinés à la première version encodée autoriseraient l'accès direct recherché.

Soit donc à constituer au départ de la version encodée du répertoire de BOENNINGHAUSEN :

_un fichier dictionnaire de niveau 1 reprenant le dictionnaire de niveau 1

_un fichier dictionnaires de niveau 2 reprenant l'ensemble des dictionnaires de niveau 2

_un fichier dictionnaires de niveau 3 reprenant l'ensemble des dictionnaires de niveau 3

_le fichier dictionnaires de niveau supérieur à 3 reprenant l'ensemble des dictionnaires de niveau supérieur à 3

sachant :

_le dictionnaire de niveau 1 formé des caractéristiques de toutes les rubriques de niveau 1

_un dictionnaire de niveau i formé des caractéristiques de toutes les rubriques de niveau i , descendantes d'une même rubrique de niveau $i-1$

entendant par caractéristiques d'une rubrique :

_de niveau 1, 2, 3 et plus

a. l'adresse dans le fichier de la version encodée, de l'intitulé attaché

b. l'adresse dans le fichier dictionnaires de niveau correspondant, du sous dictionnaire attaché

c. le nombre de rubriques du sous dictionnaire attaché

_de niveau 1, 2, 3

d. la longueur du sous dictionnaire attaché

e. son numéro d'ordre textuel parmi les rubriques de niveau 1 lorsque son niveau vaut 1, parmi les rubriques de niveau i dérivées d'une même rubrique de niveau $i-1$ lorsque son niveau i est supérieur à 1

f. le pointeur vers la clé lui correspondant.

Les dictionnaires de niveau 1, 2 et 3 ont pour organisation interne, le résultat d'une juxtaposition de deux tables :

_l'une (table 1) reprenant par rubrique concernée, dans l'ordre de leur apparition dans le répertoire de BOENNINGHAUSEN (ordre textuel), les caractéristiques a. b. c. d.

_l'autre (table 2) reprenant par rubrique concernée, dans l'ordre alphabétique des intitulés, les caractéristiques e. f.

Les deux tables ainsi juxtaposées se trouvent suivies d'une zone dite variable parce qu'elle est formée de l'ensemble des clés correspondant aux intitulés des rubriques concernées.

On parle encore de "partie rubrique" pour l'ensemble formé des deux tables, de "partie clé" pour la zone dite variable.

Soit pour un dictionnaire de niveau 1, 2 et 3, la configuration interne suivante :

table 1	table 2
zone variable	

L'ordre des dictionnaires au sein du fichier dictionnaire(s) de niveau correspondant, respecte l'ordre d'apparition des rubriques y figurant dans le répertoire de BOENNINGHAUSEN.

Les dictionnaires de niveau supérieur à 3 se groupent par ensemble formé des caractéristiques de toutes les rubriques de niveau 4 et plus dérivées d'une même rubrique de niveau 3.

L'organisation interne à l'ensemble ainsi défini, vérifie la règle disant qu'au plus tôt un dictionnaire de l'ensemble est clôturé dans le répertoire de BOENNINGHAUSEN, au plus tôt ce dictionnaire et par là, les caractéristiques a. b. c. de chaque rubrique y figurant sont reportées dans le fichier dictionnaires de niveau supérieur à 3.

L'ordre des ensembles reportés dans le fichier dictionnaires de niveau supérieur à 3 respecte l'ordre d'apparition des rubriques y figurant dans le répertoire de BOENNINGHAUSEN.

3.5.3 Implémentation de la correspondance

Principe :

parcourir séquentiellement la version encodée du répertoire de BOENNINGHAUSEN,

à la rencontre d'une rubrique de niveau N ($N \geq 1$) :

_clôturer lorsqu'ils existent, les dictionnaires(ensemble) de niveau supérieur à N, en cours de constitution, dictionnaires(ensemble) formés des caractéristiques de toutes les rubriques de niveau supérieur à N, rencontrées depuis la dernière rubrique de niveau N.

Les dictionnaires(ensemble) sont clôturés par ordre de niveau décroissant.

_ajouter les caractéristiques de la rubrique rencontrée au dictionnaire(ensemble) de niveau N, en cours de constitution.

A l'origine du principe :

l'organisation interne aux dictionnaires(ensemble) et l'impossibilité qui en résulte de reporter dans le fichier dictionnaire(s) de niveau correspondant, les caractéristiques d'une rubrique à sa rencontre dans la version encodée. D'où le décalage observé dans le principe, entre la rencontre d'une rubrique dans la version encodée et le report de ses caractéristiques dans le fichier dictionnaire(s) de niveau correspondant, report effectué à la clôture du dictionnaire l'englobant.

La rencontre autorise l'ajout des caractéristiques de la rubrique

_au dictionnaire de niveau correspondant en cours de constitution, lorsque le niveau de la rubrique est inférieur ou égal à 3.

Concrètement : le report des caractéristiques dans deux chaînes liées chacune au dictionnaire de niveau correspondant en cours de constitution;

la première reprend par rubrique appelée à constituer le dictionnaire, dans l'ordre de leur apparition au sein du répertoire de BOENNINGHAUSEN, les caractéristiques a. b. c. d. définies en 3.5.2;

la seconde reprend par rubrique appelée à constituer le dictionnaire, dans l'ordre alphabétique des intitulés, les caractéristiques e. f. définies en 3.5.2 en plus de l'intitulé même.

_à l'ensemble en cours de constitution lorsque le niveau de la rubrique est supérieur à 3.

Concrètement : le report des caractéristiques dans une chaîne liée à l'ensemble en cours de constitution;

la chaîne reprend par rubrique appelée à constituer l'ensemble, dans l'ordre de leur apparition au sein du répertoire de BOENNINGHAUSEN, les caractéristiques a. b. c. définies en 3.5.2 en plus d'un numéro d'ordre interne à l'ensemble.

Le report s'effectue à la clôture

_du dictionnaire en cours de constitution, l'englobant.

La clôture d'un dictionnaire consiste à organiser en parties rubrique et clé, les caractéristiques accumulées dans les deux chaînes qui lui sont liées, reporter le dictionnaire ainsi constitué à la suite du fichier dictionnaire(s) de niveau correspondant, mettre à jour les caractéristiques b. c. d. définies

en 3.5.2 de la rubrique directement ascendante des rubriques formant le dictionnaire constitué, rubrique associée par construction au dernier élément de la première chaîne liée au dictionnaire de niveau directement inférieur, en cours de constitution.

_de l'ensemble en cours de constitution, l'enlobant.

La clôture d'un ensemble consiste au départ de la chaîne qui lui est liée, à reporter à la suite du fichier dictionnaires de niveau supérieur à 3, un ensemble conforme aux règles définies en 3.5.2, mettre à jour les caractéristiques b, c, d, définies en 3.5.2 de la rubrique directement ascendante des rubriques formant l'ensemble ainsi constitué, rubrique associée par construction au dernier élément de la première chaîne liée au dictionnaire de niveau 3 en cours de constitution.

En termes de procédures obtenues par réduction de problèmes :

(A) la procédure principale : retranscription du principe avec prise en considération d'erreurs d'encodage au sein de la version encodée, erreurs portant sur les symboles d'entête et les valeurs de niveaux.

(B) la procédure d'initialisation :

(B.1) prologue :

_définit le fichier de la version encodée du répertoire de BOENNINGHAUSEN, l'ouvre à la lecture,

_définit le fichier dictionnaire de niveau 1, l'ouvre à l'écriture,

_définit le fichier dictionnaires de niveau 2, l'ouvre à l'écriture,

_définit le fichier dictionnaires de niveau 3, l'ouvre à l'écriture,

_définit le fichier dictionnaires de niveau supérieur à 3, l'ouvre à l'écriture,

_initialise :

le niveau courant,

le dernier niveau courant de l'ensemble en cours de constitution,

le nombre courant de rubriques du dictionnaire de niveau 1 en cours de constitution,

le nombre courant de rubriques du dictionnaire de niveau 2 en cours de constitution,

le nombre courant de rubriques du dictionnaire de niveau 3 en cours de constitution,

le nombre courant de rubriques de niveau 4 de l'ensemble en cours de constitution,

le nombre courant de rubriques de l'ensemble en cours de constitution, reportées à la suite du fichier dictionnaires de niveau > 3

l'indice attaché au fichier de la version encodée,

l'indice attaché au fichier dictionnaires de niveau 2,

l'indice attaché au fichier dictionnaires de niveau 3,

l'indice attaché au fichier dictionnaires de niveau > 3,

l'indice attaché à l'ensemble en cours de constitution

_crée :

l'élément de tête de la 1ère chaîne liée au dictionnaire de niveau 1 en cours de constitution,

l'élément de tête de la 1ère chaîne liée au dictionnaire de niveau 2 en cours de constitution,

l'élément de tête de la 1ère chaîne liée au dictionnaire de niveau 3 en cours de constitution,

l'élément de tête de la 2ème chaîne liée au dictionnaire de niveau 1 en cours de constitution,

l'élément de tête de la 2ème chaîne liée au dictionnaire de niveau 2 en cours de constitution,

l'élément de tête de la 2ème chaîne liée au dictionnaire de niveau 3 en cours de constitution,

l'élément de tête de la chaîne liée à l'ensemble en cours de constitution.

(C) la procédure de terminaison :

(C.1) épilogue :

_clôture par ordre de niveau décroissant, les dictionnaires(ensemble) en cours de constitution,

_ferme les quatre fichiers définis,

_provoque l'arrêt du programme.

(D) les procédures de clôture :

(D.1) copier dic1 :

_établit les parties rubrique et clé du dictionnaire de niveau 1 en cours de constitution,

_reporte dans le fichier dictionnaire de niveau 1, le dictionnaire ainsi constitué,

_fait afficher les longueur et nombre de rubriques du dictionnaire ainsi constitué.

(D.2) copier dic2 :

_établit les parties rubrique et clé du dictionnaire de niveau 2 en cours de constitution,

_reporte à la suite du fichier dictionnaires de niveau 2, le dictionnaire ainsi constitué,

_met à jour les caractéristiques b. c. d. définies en 3.5.2 de la rubrique de niveau 1 directement ascendante des rubriques formant le dictionnaire ici constitué,

_met à jour l'indice attaché au fichier dictionnaires de niveau 2,

_réinitialise les 2 chaînes et le nombre courant de rubriques, liés au dictionnaire de niveau 2 en cours de constitution.

(D.3) copier dic3 :

_établit les parties rubrique et clé du dictionnaire de niveau 3 en cours de constitution,

_reporte à la suite du fichier dictionnaires de niveau 3, le dictionnaire ainsi constitué,

_met à jour les caractéristiques b. c. d. définies en 3.5.2 de la rubrique de niveau 2 directement ascendante des rubriques formant le dictionnaire ici constitué,

_met à jour l'indice attaché au fichier dictionnaires de niveau 3,

_réinitialise les 2 chaînes et le nombre courant de rubriques, liés au dictionnaire de niveau 3 en cours de constitution.

(D.4) copier diction

_reporte à la suite du fichier dictionnaire de niveau supérieur à 3, par ordre de niveau décroissant, tous les dictionnaires non encore reportés de l'ensemble en cours de constitution,

_met à jour les caractéristiques b. c. d. définies en 3.5.2 de la rubrique de niveau 3 directement ascendante des rubriques formant l'ensemble ici constitué,

_met à jour l'indice attaché au fichier dictionnaires de niveau > 3,

_réinitialise la chaîne, le dernier niveau courant, l'indice, le nombre courant de rubriques de niveau 4, le nombre courant de rubriques reportées à la suite du fichier dictionnaires de niveau > 3, liés à l'ensemble en cours de constitution.

(D.5) établir clé :

_établit au départ de la chaîne d'élément de tête passé en argument, la partie clé correspondante,
_la reporte dans l'endroit prévu à cet effet,
_met à jour la caractéristique f. définie en 3.5.2 de chacun des éléments de la chaîne d'élément de tête passé en argument.

(D.6) cas2 :

_établit au départ des trois intitulés passés en argument, respectivement : l'intitulé précédent, l'intitulé, l'intitulé suivant, la clé correspondant à l'intitulé,
_la reporte suivie du caractère nul à la suite des clés déjà établies pour le dictionnaire à constituer, dans l'endroit prévu à cet effet.

(D.7) cas1 :

_établit au départ des deux intitulés passés en argument, respectivement : l'intitulé et l'intitulé suivant, la clé correspondant à l'intitulé,
_la reporte suivie du caractère nul à la suite des clés déjà établies pour le dictionnaire à constituer, dans l'endroit prévu à cet effet.

(D.8) copier rub :

_juxtapose les caractéristiques a. b. c. d. e. f. définies en 3.5.2 des deux chaînes d'élément de tête passés en argument,
_reporte la partie rubrique ainsi formée à la suite du fichier dictionnaire de niveau passé en argument.

(D.9) copier clé :

_reporte la partie clé reprise en l'endroit prévu à cet effet, à la suite du fichier dictionnaire de niveau passé en argument.

(D.10) mise à neuf123 :

_libère les deux chaînes d'élément de tête passés en argument, de tous les éléments autres que l'élément de tête,
_réinitialise le nombre courant de rubriques passé en argument.

(D.11) mise à neufn :

_libère la chaîne liée à l'ensemble en cours de constitution de tous les éléments autres que l'élément de tête,
_réinitialise le nombre courant de rubrique de niveau 4, le nombre courant de rubriques reportées à la suite du fichier dictionnaires de niveau > 3, l'indice et le dernier niveau courant, attachés à l'ensemble en cours de constitution.

(D.12) mise à jour :

_lorsque la valeur passée en argument est nulle,
fait afficher les longueur et nombre de rubriques du dictionnaire
de niveau 1

_lorsque la valeur passée en argument est > 0 ,
met à jour les caractéristiques b. c. d. définies en 3.5.2 de la
rubrique directement ascendante des rubriques formant le dernier
dictionnaire(ensemble) de niveau égal à la valeur passée en
argument plus un, constitué,
met à jour l'indice attaché au fichier dictionnaire de niveau égal
à la valeur passée en argument plus un.

(E) les procédures d'ajout :

(E.1) traiter intitulé123 :

_crée un élément reprenant les caractéristiques a. b. c. d.
définies en 3.5.2 pour l'insérer au bout de la première chaîne
d'élément de tête passé en argument,

_crée un élément reprenant les caractéristiques e. f. définies en
3.5.2 en plus de l'intitulé même pour l'insérer au sein de la
deuxième chaîne d'élément de tête passé en argument,
conformément à l'ordre alphabétique des intitulés,

_incrémente d'une unité le nombre courant de rubriques passé en
argument.

(E.2) traiter intitulen :

_si le niveau courant (niveau de la rubrique courante) est
inférieur au dernier niveau courant de l'ensemble en cours de
constitution :

reporte à la suite du fichier dictionnaires de niveau > 3 , par
ordre de niveau décroissant, tous les dictionnaire de niveau
supérieur au niveau courant, non encore reportés de l'ensemble en
cours de constitution.

_crée un élément reprenant les caractéristiques a. b. c. définies en
3.5.2 en plus de l'indice interne à l'ensemble en cours de
constitution, pour l'insérer au bout de la chaîne liée à
l'ensemble en cours de constitution,

_incrémente d'une unité le nombre courant de rubriques de niveau 4
(s'il y a lieu) et l'indice, attachés à l'ensemble en cours de
constitution,

_met à jour le dernier niveau courant.

(E.3) sup :

_compare deux intitulés passés en argument,
renvoie 1 lorsque le premier intitulé est alphabétiquement
supérieur au second,
renvoie 0 lorsque le premier intitulé est alphabétiquement
inférieur au second,
signale un intitulé double et provoque l'arrêt immédiat lorsque les
deux intitulés sont égaux ou inclus l'un dans l'autre.

(F) les procédures d'erreurs :

(F.1) erreur1 :

_signale et localise à l'écran une rupture de niveaux,
_renvoie la valeur corrigée, elle-même obtenue par interaction.

(F.2) erreur2 :

_signale et localise à l'écran un symbole d'entête inconnu,
_provoque l'arrêt du programme.

(F.3) erreur3 :

_signale et localise à l'écran un intitulé double,
_provoque l'arrêt du programme.

(F.4) affichage hierarchise :

_affiche à l'écran sur la (9+i)ème ligne, depuis la 9ème colonne,
l'intitulé de la rubrique de niveau i, dernière relevée dans le
parcours séquentiel du fichier de la version encodée du répertoire
de BOENNINGHAUSEN, i valant successivement 1, 2, ...niveau courant.

(F.5) affichage :

_affiche à l'écran sur la (9+i)ème ligne, depuis la ième colonne
(avec i passé en argument), l'intitulé lu dans le fichier de la
version encodée du répertoire de BOENNINGHAUSEN, à l'adresse passée
en argument.

(G) les procédures auxiliaires :

(G.1) copier sup :

_reporte par ordre de niveau décroissant, à la suite du fichier
dictionnaires de niveau supérieur à 3, les dictionnaires de niveau
supérieur à la valeur passée en argument, non encore reportés, de
l'ensemble en cours de constitution,
_met à jour les caractéristiques b. c. définies en 3.5.2 des
rubriques directement ascendantes de chaque dictionnaire reporté,
_met à jour le nombre de rubriques de l'ensemble en cours de
constitution, reportées à la suite du fichier dictionnaires de
niveau > 3.

(G.2) get chr :

_incréménte d'une unité l'indice attaché au fichier de la version encodée du répertoire de BOENNINGHAUSEN,

_lit et renvoie le caractère suivant dans le fichier de la version encodée du répertoire de BOENNINGHAUSEN.

En résumé, le programme formé des procédures ici décrites établit au départ du fichier de la version encodée du répertoire de BOENNINGHAUSEN décrit en 3.4.2, les quatre fichiers dictionnaires de niveau 1, 2, 3 et plus décrits en 3.5.2, affiche à l'écran les longueur et nombre de rubriques du dictionnaire de niveau 1.

S'il repère en cours d'établissement un intitulé double (traduisez : parmi les rubriques de niveau i dérivées d'une même rubrique de niveau i-1, l'intitulé d'une rubrique identique ou inclu à l'intitulé d'une deuxième), un symbole d'entête inconnu, signale et localise à l'écran l'incident pour s'arrêter aussitôt.

S'il repère en cours d'établissement une rupture de niveaux (traduisez : le niveau d'une rubrique supérieur de plus d'une unité au dernier niveau relevé), en demande la correction.

3.6 MISE A JOUR

La mise à jour concerne les deux versions encodées du répertoire de KENT; la deuxième se trouvant dérivée de la première par éclatement, la mise à jour à concevoir portera sur la première version encodée, elle-même alors éclatée en la deuxième version encodée, par construction mise à jour.

Soit donc à mettre à jour la première version encodée du répertoire de KENT par

_insertion de groupes de rubriques, traduisez : insertion groupée de rubriques à la fois adjacentes et de même niveau au sein du répertoire de BOENNINGHAUSEN, en un endroit donné du répertoire de KENT.

Le groupe est inséré avec ou sans l'entière des rubriques dérivées des rubriques le constituant.

_fusion de paires de rubriques, traduisez : insertion des remèdes attachés à une rubrique du répertoire de BOENNINGHAUSEN, parmi les remèdes attachés à une rubrique du répertoire de KENT, étant entendu que

l'accès aux répertoires qu'il soit de BOENNINGHAUSEN pour la délimitation d'un groupe, la localisation d'une rubrique, de KENT pour la localisation de l'insertion, s'effectue de façon hiérarchisée conformément à la structure des deux répertoires (on parle encore d'accès direct),

la localisation de l'insertion ne subit aucune restriction,

l'introduction des données relatives aux éléments à insérer ou fusionner s'effectue en une voire plusieurs sessions au gré de l'utilisateur.

Les données à communiquer par groupe de rubriques à insérer auront trait à :

_la délimitation dans le répertoire de BOENNINGHAUSEN du groupe à insérer;

en supposant le groupe formé de rubriques de niveau i ($i \geq 1$), seront successivement demandées les rubriques de niveau 1, 2, ... $i-1$ ascendantes du groupe ainsi que les première et dernière rubriques de niveau i du groupe.

_la localisation de l'insertion dans le répertoire de KENT étant entendu q'un groupe peut s'insérer de trois façons possibles :

.comme dérivé d'une rubrique donnée, immédiatement avant une sous-rubrique donnée;

en supposant la rubrique donnée de niveau j ($j \geq 0$), seront successivement demandées les rubriques de niveau 1, 2, ... $j-1$ ascendantes de la rubrique donnée ainsi que la rubrique et la sous-rubrique données;

.comme dérivé d'une rubrique donnée, immédiatement après la dernière sous-rubrique;

en supposant la rubrique donnée de niveau j ($j \geq 0$), seront successivement demandées les rubriques de niveau 1, 2, ... $j-1$ ascendantes de la rubrique donnée ainsi que la rubrique donnée;

.comme dérivé d'une rubrique donnée, n'ayant aucune sous-rubrique;

en supposant la rubrique donnée de niveau j ($j \geq 0$), seront successivement demandées les rubriques de niveau 1, 2 ... $j-1$ ascendantes de la rubrique donnée ainsi que la rubrique donnée.

_la mise en forme ou conversion des rubriques à insérer;

sera demandé s'il faut ou non prendre en considération l'entièreté des rubriques dérivées des rubriques constituant le groupe;

sera demandé à chaque changement d'écriture relevé dans les intitulés de rubriques à insérer, s'il convient ou non de scinder;

sera offerte la possibilité de modifier le niveau d'une rubrique à insérer lorsque se sera produit au niveau directement inférieur soit une scission soit une modification de niveaux;

sera demandée la correspondance de toute abréviation relevée dans les intitulés de rubriques à insérer;

sera demandée la correction de tout remède relevé parmi les remèdes attachés aux rubriques à insérer lorsqu'il s'avèrera inconnu.

Les données à communiquer par paire de rubriques à fusionner auront trait à :

_la localisation dans le répertoire de BOENNINGHAUSEN de la rubrique à fusionner;

en supposant la rubrique de niveau i ($i \geq 1$), seront

successivement demandées les rubriques de niveau 1, 2 ... i-1 ascendantes de la rubrique à fusionner ainsi que la rubrique à fusionner;

_la localisation dans le répertoire de KENT de la rubrique à fusionner;

en supposant la rubrique de niveau j ($j \geq 1$), seront successivement demandées les rubriques de niveau 1, 2 ... j-1 ascendantes de la rubrique à fusionner ainsi que la rubrique à fusionner;

_la correction des remèdes;

sera demandée la correction de tout remède relevé parmi les remèdes attachés à la rubrique du répertoire de BOENNINGHAUSEN, à fusionner lorsqu'il s'avèrera inconnu.

Soit un nombre particulièrement élevé de données à communiquer tant pour l'insertion d'un groupe de rubriques que la fusion d'une paire de rubriques.

D'où l'accent spécialement mis sur la commodité à l'emploi de la mise à jour à concevoir, commodité ici atteinte par l'application de trois mesures consistant à :

_autoriser le travail par session, concrètement donner à l'utilisateur la possibilité de répartir l'introduction des données sur plusieurs sessions, la mise à jour s'effectuant lors de la dernière session, elle-même décidée par l'utilisateur.

_séparer autant que faire se peut la mise à jour proprement dite de l'introduction des données, concrètement faire en sorte que la mise à jour se ramène à une juxtaposition de pièces tantôt prélevées dans le répertoire de KENT non mis à jour, tantôt prélevées dans le répertoire de BOENNINGHAUSEN mis en forme.

_récupérer autant que faire se peut les erreurs, en particulier celles effectuées à l'encodage du répertoire de BOENNINGHAUSEN, au sein des différents Bloc-remèdes;

autoriser en d'autres termes, la correction de tout remède incorrectement écrit.

.celles effectuées à l'introduction des données;

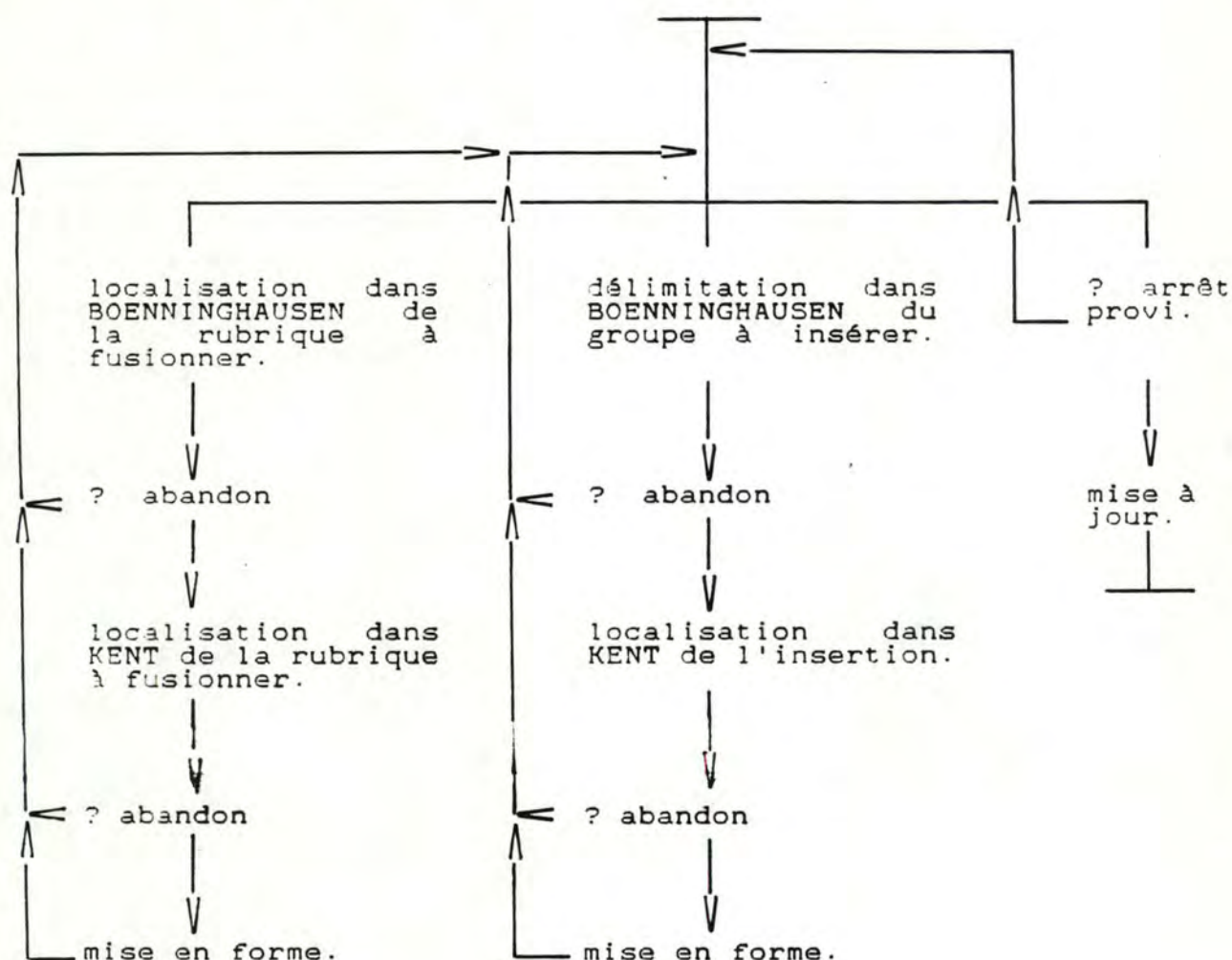
permettre en d'autres termes, la suppression de l'entièreté des données communiquées depuis la dernière mise à jour;

permettre la suppression des données communiquées dans le cadre de l'insertion en cours par abandon lors de la délimitation du groupe à insérer ou lors de la localisation de l'insertion;

permettre la suppression des données communiquées dans le cadre de la fusion en cours par abandon lors de la localisation dans le répertoire de BOENNINGHAUSEN ou lors de la localisation dans le répertoire de KENT;

permettre la modification des données communiquées dans le cadre d'une délimitation, d'une localisation aussi longtemps que la délimitation, la localisation ne correspond pas à celle souhaitée.

Scénario imaginé pour la mise à jour à concevoir :



3.6.1 Principe retenu pour la mise à jour de la première version encodée du répertoire de KENT

Au fur et à mesure de l'introduction des données :
pour chaque groupe de rubriques à insérer (paire de rubriques à fusionner),

_mettre en forme le groupe de rubriques à insérer (la liste des remèdes attachés à la rubrique du répertoire de BOENNINGHAUSEN à fusionner),

le(la) reporter à la suite des groupes(listes) mis en forme depuis la dernière mise à jour de la première version encodée du répertoire de KENT,

_établir l'information nécessaire à l'insertion dans la première version encodée du répertoire de KENT, du groupe(de la liste) mis(e) en forme,

la reporter parmi les informations relatives à chaque groupe(chaque liste) mis(e) en forme depuis la dernière mise à jour de la première version encodée du répertoire de KENT, report conforme à l'ordre textuel des insertions à opérer dans la première version encodée du répertoire de KENT.
L'introduction des données clôturée :

_juxtaposer sur base des informations relatives à chaque groupe(chaque liste) mis(e) en forme, la première version encodée du répertoire de KENT avec les groupes(listes) mis en forme, ainsi constituer la première version encodée du répertoire de KENT, mise à jour.

Entendant par :

_mise en forme d'un groupe :

la mise en forme de tout intitulé s'y trouvant par l'ajout des caractères annonçant et terminant tout intitulé de la première version encodée du répertoire de KENT, l'établissement du niveau qu'aura l'intitulé dans la première version encodée du répertoire de KENT mise à jour et s'il y lieu, la modification de niveaux établis, la substitution d'abréviations et la scission,

la mise en forme de tout Bloc-remèdes s'y trouvant par l'ajout des caractères annonçant et terminant tout Bloc-remèdes de la première version encodée du répertoire de KENT, la substitution à chaque remède de sa correspondance codée dans la première version encodée du répertoire de KENT et la conversion des

degrés attachés.

_mise en forme de la liste des remèdes attachés à une rubrique du répertoire de BOENNINGHAUSEN à fusionner :

la substitution à chaque remède de sa correspondance dans la version originale du répertoire de KENT et la conversion des degrés attachés.

Le fait de ne pas substituer à chaque remède sa correspondance codée dans la première version encodée du répertoire de KENT s'explique par la différence relevée entre l'ordonnancement alphabétique des codes et l'ordonnancement de leur correspondance dans la version originale du répertoire de KENT; les remèdes se trouvant codés au fur et à mesure de leur apparition dans la version originale du répertoire de KENT, l'ordonnancement des codes de la première version encodée du répertoire de KENT n'est donc pas exactement alphabétique; d'où la nécessité pour l'ordonnancement des remèdes issus de deux listes à fusionner de passer par les correspondances dans la version originale du répertoire de KENT.

_information nécessaire à l'insertion d'un groupe mis en forme
a. le niveau qu'aura le groupe dans la première version encodée du répertoire de KENT mise à jour

b. le type (de l'insertion) du groupe :

inséré comme dérivé d'une rubrique donnée, immédiatement avant une sous-rubrique donnée, le groupe sera dit E(lse)

inséré comme dérivé d'une rubrique donnée, immédiatement après la dernière sous-rubrique, le type sera dit F(in-dic)

inséré comme dérivé d'une rubrique donnée, n'ayant aucune sous-rubrique, le type sera dit N(ew-dic)

c. la localisation de la rubrique dite d'insertion, rubrique de la première version encodée du répertoire de KENT, par définition condamnée à suivre directement le groupe dans la première version encodée du répertoire de KENT mise à jour

d. la localisation du groupe parmi l'entièreté des groupes(listes) mis en forme.

L'information se trouve reportée parmi les informations relatives à chaque groupe(chaque liste) mis(e) en forme depuis la dernière mise à jour de la première version encodée du répertoire de KENT, report conforme à l'ordre textuel des rubriques d'insertion dans la première version encodée du répertoire de KENT.

Lorsque deux voire plusieurs groupes présentent la même rubrique d'insertion : l'information relative au groupe de type N précède l'information relative au groupe de type F qui elle-même précède l'information relative au groupe de type E; lorsqu'en outre, les deux groupes s'avèrent

de type N :

l'information relative au groupe le dernier introduit précède l'information relative à l'autre;

de type F :

l'information relative au groupe de niveau le plus élevé précède l'information relative à l'autre;

de type E :

l'information relative au groupe le dernier introduit précède l'information relative à l'autre.

_information nécessaire à l'insertion d'une liste mise en forme

- a. la localisation de la rubrique dite d'insertion, rubrique de la première version encodée du répertoire de KENT par définition confondue à la rubrique désignée pour être fusionnée
- b. la localisation de la liste parmi l'entièreté des groupes(listes) mis en forme.

Lorsqu'un groupe et une liste présentent la même rubrique d'insertion, du fait des deux définitions de la rubrique d'insertion, l'information relative au groupe précède l'information relative à la liste.

_juxtaposition sur base d'informations relatives à chaque groupe(liste) mis en forme ... :

prise en considération dans l'ordre de leur apparition, de chaque information pour effectuer successivement :

1. la reprise de la première version encodée du répertoire de KENT depuis l'endroit atteint jusque la rubrique d'insertion (exclue) mentionnée dans l'information
 2. la reprise du groupe mis en forme lorsque l'information a trait à un groupe,
- la reprise de l'intitulé de la rubrique d'insertion et du résultat sous forme codée, de la fusion des remèdes de la liste mise en forme avec s'ils existent, les remèdes de la rubrique d'insertion, lorsque l'information a trait à une liste.

3.6.2 Implémentation

Chaque groupe(liste) mis(e) en forme est reporté(e) à la suite d'un fichier dit des groupes(listes) mis en forme.

L'information qu'elle soit relative à un groupe mis en forme, à une liste mise en forme, comporte quatre points :

- _le niveau (fictif dans le cas d'une liste)
- _le type de l'insertion (N, F, ou E dans le cas d'un groupe, * dans le cas d'une liste)
- _la localisation de l'intitulé de la rubrique d'insertion dans le fichier des rubriques
- _la localisation du groupe(de la liste) dans le fichier des groupes(listes) mis en forme.

L'information quelle soit relative à un groupe mis en forme, à une liste mise en forme, se trouve reprise dans l'élément d'une chaîne dite d'information, chaîne organisée de façon à respecter l'ordre textuel des intitulés des rubriques d'insertion dans le fichier des rubriques.

En fin de session autre que terminale :

l'information reprise dans chaque élément de la chaîne d'information , la valeur de l'indice attaché au fichier des groupes(listes) mis en forme et la valeur du niveau le plus haut dans l'entièreté des groupes mis en forme se trouvent successivement reportés dans un fichier dit d'information.

En fin de session terminale :

le contenu du fichier d'information se trouve supprimé,
la valeur réinitialisée de l'indice attaché au fichier des groupes(listes) mis en forme et celle réinitialisée du niveau le plus haut dans l'entièreté des groupes mis en forme sont reportées dans le fichier d'information,
le contenu du fichier des groupes(listes) mis en forme se trouve supprimé.

En début de session (terminale et autre)..:

lorsqu'il se trouve de l'information dans le fichier d'information, chaque information se voit reprise dans un élément créé à la suite de la chaîne d'information,
la valeur de l'indice attaché au fichier des groupes(listes) mis en forme se trouve retenu,
la valeur du niveau le plus haut dans l'entièreté des groupes mis en forme se trouve retenu.

Le fichier dit des remèdes et correspondances permet d'établir la correspondance dans la version originale du répertoire de KENT et la correspondance codée dans la première version encodée du répertoire de KENT, de tout remède issu de la version originale du répertoire de BOENNINGHAUSEN.

Le fichier reprend par ligne, le remède dans la version originale du répertoire de BOENNINGHAUSEN * la correspondance dans la version originale du répertoire de KENT * la correspondance codée dans la première version encodée du répertoire de KENT. Le fichier se trouve trié par ordre alphabétique des correspondances dans la version originale du répertoire de KENT.

Par soucis d'efficacité dans la recherche des correspondances, trois chaînes sont établies au départ du fichier des remèdes et correspondances.

principe de leur établissement : lecture séquentielle du fichier des remèdes et correspondances, à la rencontre d'un premier caractère de (remède dans la version originale du répertoire de BOENNINGHAUSEN, correspondance dans la version originale du répertoire de KENT, correspondance codée dans la première version encodée du répertoire de KENT), différent du premier caractère de (remède ..., correspondance..., correspondance...) déjà rencontrés, ajouter à la chaîne dite (boe>kent, kent>abre, abre>kent) un nouvel élément reprenant l'adresse au sein du fichier des remèdes et correspondances ainsi que la valeur du premier caractère considéré.

L'accès au répertoire de BOENNINGHAUSEN pour la délimitation d'un groupe à insérer, la localisation d'une rubrique à fusionner, au répertoire de KENT pour la localisation d'une insertion s'effectue de façon hiérarchisée(directe).

Les caractéristiques de l'accès en cours au répertoire de BOENNINGHAUSEN se trouvent reprises dans un tableau dit tableau_b, formé par niveau autorisé dans le répertoire de BOENNINGHAUSEN, des caractéristiques : adresse au sein du fichier-dictionnaire(s) l'enfermant, longueur et nombre de rubriques de chaque dictionnaire traversé et de la caractéristique : adresse dans la version encodée du répertoire de BOENNINGHAUSEN, de chaque rubrique le jalonnant.

Les caractéristiques de l'accès en cours au répertoire de KENT se trouvent reprises dans un tableau dit tableau_k, formé par niveau autorisé par le répertoire de KENT, des caractéristiques : adresse au sein du fichier-dictionnaire(s) l'enfermant, longueur et nombre de rubriques, de chaque dictionnaire traversé et de la caractéristique : adresse au sein du fichier des rubriques, de chaque rubrique le jalonnant.

Soit en termes de procédures, obtenues par réduction de problèmes :

(A) la procédure principale :
retranscription du scénario.

(B) la procédure d'initialisation :

(B.1) prologue :

_définit le fichier de la version encodée du répertoire de BOENNINGHAUSEN, l'ouvre à la lecture,

_définit le fichier-dictionnaire de niveau 1 du répertoire de BOENNINGHAUSEN, l'ouvre à la lecture,

_définit le fichier-dictionnaires de niveau 2 du répertoire de BOENNINGHAUSEN, l'ouvre à la lecture,

_définit le fichier-dictionnaires de niveau 3 du répertoire de BOENNINGHAUSEN, l'ouvre à la lecture,

_définit le fichier-dictionnaires de niveau > 3 du répertoire de BOENNINGHAUSEN, l'ouvre à la lecture,

_définit le fichier des remèdes et correspondances, l'ouvre à la lecture,

_définit le fichier de la première version encodée du répertoire de KENT, l'ouvre à la lecture,

_définit le fichier-dictionnaire de niveau 1 du répertoire de KENT, l'ouvre à la lecture,

_définit le fichier-dictionnaires de niveau 2 du répertoire de KENT, l'ouvre à la lecture,

_définit le fichier-dictionnaires de niveau 3 du répertoire de KENT, l'ouvre à la lecture,

_définit le fichier-dictionnaires de niveau > 3 du répertoire de KENT, l'ouvre à la lecture,

_définit le fichier des rubriques du répertoire de KENT, l'ouvre à la lecture,

_définit le fichier la première version encodée du répertoire de KENT mise à jour, l'ouvre à l'écriture

_définit le fichier d'information, l'ouvre à la lecture,

_définit le fichier des groupes(listes) mis en forme, l'ouvre en mode append

_crée l'élément de tête de la chaîne d'information,

lorsqu'il se trouve repris de l'information dans le fichier d'information,

crée par information, à la suite de la chaîne d'information, un élément pour reprendre l'information,

retient la valeur de l'indice attaché au fichier des groupes(listes) mis en forme,

retient la valeur du niveau le plus haut dans l'entièreté des groupes(listes) mis en forme,

ferme le fichier d'information,

_établit les longueurs :

du fichier de la version encodée du répertoire de BOENNINGHAUSEN

du fichier des rubriques

du fichier de la première version encodée du répertoire de KENT

du fichier des remèdes et correspondances,

_établit les 3 chaînes : boe>kent, abre>kent, kent>abre

_initialise tableau_b, tableau_k

_crée :

l'élément de tête d'une chaîne dite chaîne_1, formée d'éléments reprenant chacun le numéro d'ordre d'une rubrique retenue dans la recherche de rubriques en cours,

l'élément de tête d'une chaîne dite chaîne_2, formée d'éléments-pointeurs vers les (n*M)èmes éléments de chaîne_1 avec pour n, les valeurs 0, 1, 2 ... et pour M, le nombre maximum autorisé d'intitulés affichés dans le cas d'un affichage multiple,

l'élément de tête d'une chaîne liée aux abréviations relevées dans la groupe à mettre en forme, formée d'éléments reprenant chacun l'abréviation et sa correspondance, obtenue par interaction,

l'élément de tête d'une chaîne liée aux listes de remèdes à fusionner, chaîne formée d'éléments reprenant chacun par remède, sa correspondance dans la version originale du répertoire de KENT, le degré affecté et la référence attachée.

(C) la procédure de terminaison :

(C.1) épilogue :

_lorsque la valeur passée en argument est 1 :
 ferme tous les fichiers définis encore ouverts,
 supprime le contenu du fichier d'information,
 y reporte l'information reprise dans chaque élément de la chaîne d'information,
 y reporte la valeur atteinte par l'indice attaché au fichier des groupes(listes) mis en forme,
 y reporte la valeur du niveau le plus haut dans l'entièreté des groupes mis en forme,
 ferme le fichier d'information.

_lorsque la valeur passée en argument est 2 :
 ferme le fichier des groupes(listes) mis en forme.

_lorsque la valeur passée en argument est 3 :
 ferme tous les fichiers définis encore ouverts,
 supprime le contenu du fichier des groupes(listes) mis en forme,
 supprime le contenu du fichier d'information,
 y reporte la valeur réinitialisée de l'indice attaché au fichier des groupes(listes) mis en forme,
 y reporte la valeur réinitialisée du niveau le plus haut dans l'entièreté des groupes mis en forme,
 ferme le fichier d'information.

(D) les procédures d'introduction des données :

(D.1) définir deb :

_réinitialise le niveau BOENNINGHAUSEN courant,
 _autorise l'accès direct au répertoire de BOENNINGHAUSEN pour la définition de la première rubrique du groupe à insérer,
 _lorsqu'il y a définition :
 retient le niveau BOENNINGHAUSEN du groupe à insérer,
 retient l'adresse au sein de la version encodée du répertoire de BOENNINGHAUSEN, de la rubrique définie première,
 renvoie 1,
 lorsqu'il n'y a pas définition :
 renvoie 0.

(D.2) définir fin :

_autorise l'accès aux seules rubriques du répertoire de BOENNINGHAUSEN, de caractéristiques reprises dans le dictionnaire, de caractéristiques elles-mêmes reprises au niveau BOENNINGHAUSEN courant de tableau_b, pour la définition de la

dernière rubrique du groupe à insérer,

_lorsqu'il y a définition et lorsque la rubrique définie dernière ne précède pas la rubrique définie première dans la version encodée du répertoire de BOENNINGHAUSEN :

retient l'intitulé de la rubrique définie dernière,
renvoie 1,

lorsqu'il n'y a pas définition ou lorsque la rubrique définie dernière précède la rubrique définie première dans la version encodée du répertoire de BOENNINGHAUSEN :

renvoie 0.

(D.3) définir insert :

_réinitialise le niveau KENT courant,

_autorise l'accès direct au répertoire de KENT pour la localisation de l'insertion du groupe à insérer,

_lorsqu'il y a localisation :

retient le niveau qu'aura le groupe dans la première version encodée du répertoire de KENT mise à jour,

établit le type du groupe (F, N, E),

renvoie 1,

lorsqu'il n'y a pas localisation :

renvoie 0.

(D.4) définir liste1 :

_réinitialise le niveau BOENNINGHAUSEN courant,

_autorise l'accès direct au répertoire de BOENNINGHAUSEN pour la définition de la rubrique à fusionner,

_lorsqu'il y a définition et lorsqu'une liste de remèdes se trouve attachée à la rubrique définie :

retient l'adresse au sein de la version encodée du répertoire de BOENNINGHAUSEN, de la rubrique définie,

renvoie 1,

lorsqu'il n'y a pas définition ou lorsqu'aucune liste de remèdes se trouve attachée à la rubrique définie :

renvoie 0.

(D.5) définir liste2 :

_réinitialise le niveau KENT courant,

_autorise l'accès direct au répertoire de KENT pour la définition de la rubrique à fusionner,

_lorsqu'il y a définition :

pose le type égal à *,

renvoie 1

lorsqu'il n'y a pas définition :

renvoie 0.

(D.6) mode n :

_autorise de façon interactive, la modification du niveau BOENNINGHAUSEN courant lorsque le mode a pour valeur courante 1, 2 (du niveau KENT courant lorsque le mode a pour valeur courante 3), par décrémentation jusqu'au plus 1, incrémentation jusqu'au plus le niveau de la dernière rubrique, de caractéristiques reportées dans tableau_b lorsque la mode a pour valeur courante 1, 2 (dans tableau_k lorsque le mode a pour valeur courante 3).

(D.7) mode i1 :

_fixe la valeur courante du mode à 1,
_observe le cycle ici décrit avec arrêt décidé de façon interactive et possibilité de modifier le niveau BOENNINGHAUSEN courant :

.demande de façon interactive, les premiers caractères significatifs de la rubrique de niveau BOENNINGHAUSEN courant,
.lorsqu'une rubrique de caractéristiques reprises dans le dictionnaire, de caractéristiques elles-mêmes reprises au niveau BOENNINGHAUSEN courant de tableau_b, s'avère y correspondre soit d'office soit par sélection :

met à jour tableau_b, des caractéristiques de la rubrique,
affiche à l'écran, l'intitulé de la rubrique,
incrémente d'une unité le niveau BOENNINGHAUSEN courant,
lorsqu'aucune rubrique de caractéristiques reprises dans le dictionnaire, de caractéristiques elles-mêmes reprises au niveau BOENNINGHAUSEN courant de tableau_b, n'y correspond :

affiche à l'écran un message de non-existence.

L'arrêt directement suivant la production d'un message de non-existence ou la modification du niveau BOENNINGHAUSEN courant est interprété comme un abandon de la définition affichée :

provoque le renvoi de 0.

L'arrêt "autre" est interprété comme un acquiescement de la définition affichée :

décrémente d'une unité le niveau BOENNINGHAUSEN courant,
provoque le renvoi de 1.

(D.8) mode i2 :

_fixe la valeur courante du mode à 2,
_observe le cycle ici décrit jusqu'à arrêt décidé de façon interactive .

.demande de façon interactive, les premiers caractères significatifs de la rubrique de niveau BOENNINGHAUSEN courant, lorsqu'une rubrique de caractéristiques reprises dans le dictionnaire de caractéristiques elles-mêmes reprises au niveau BOENNINGHAUSEN courant de tableau_b, s'avère y correspondre soit d'office soit par sélection :

met à jour tableau_b, des caractéristiques de la rubrique, affiche à l'écran, l'intitulé de la rubrique, lorsqu'aucune rubrique de caractéristiques reprises dans le dictionnaire de caractéristiques elles-mêmes reprises au niveau BOENNINGHAUSEN courant de tableau_b, n'y correspond : affiche à l'écran, un message de non-existence.

L'arrêt directement suivant la production d'un message de non-existence est interprété comme un abandon de la définition affichée et provoque le renvoi de 0.

L'arrêt "autre" est interprété comme un acquiescement de la définition affichée et provoque le renvoi de 1.

(D.9) mode i3 :

_fixe la valeur courante du mode à 3,
_observe le cycle ici décrit avec arrêt décidé de façon interactive et possibilité de modifier le niveau KENT courant :
.demande de façon interactive, les premiers caractères significatifs de la rubrique de niveau KENT courant, lorsqu'une rubrique de caractéristiques reprises dans le dictionnaire de caractéristiques elles-mêmes reprises au niveau KENT courant de tableau_k, s'avère y correspondre soit d'office, soit par sélection :

met à jour tableau_k, des caractéristiques de la rubrique, affiche à l'écran, l'intitulé de la rubrique, incrémente d'une unité le niveau KENT courant, lorsqu'aucune rubrique de caractéristiques reprises dans le dictionnaire de caractéristiques elles-mêmes reprises au niveau KENT courant de tableau_k, n'y correspond : affiche à l'écran un message de non-existence.

L'arrêt directement suivant la production d'un message de non-existence ou la modification du niveau-KENT courant, est interprété comme un abandon de la définition affichée : renvoie 0.

L'arrêt "autre" est interprété comme un acquiescement de la définition affichée :

décrémente d'une unité le niveau KENT courant,
renvoie 1.

(D.10) chercher i12 :

_fait rechercher parmi les rubriques de caractéristiques
reprises dans le dictionnaire de caractéristiques elles-mêmes
reprises au niveau BOENNINGHAUSEN courant de tableau_b, celles
dont l'intitulé est égal ou inclut l'intitulé passé en argument,
_lorsqu'aucune rubrique n'est trouvée :

affiche à l'écran, un message de non-existence,
renvoie 0,

lorsqu'une seule rubrique est trouvée :

met à jour tableau_b, des caractéristiques de la rubrique,
affiche à l'écran, l'intitulé de la rubrique,
renvoie 1,

lorsque plusieurs rubriques sont trouvées :

en demande la sélection,

met à jour tableau_b, des caractéristiques de la rubrique
sélectionnée,

affiche à l'écran, la rubrique sélectionnée,
renvoie 1.

(D.11) chercher i3 :

_lorsque l'intitulé passé en argument est le caractère "." :
lorsque la longueur du dictionnaire de caractéristiques reprises
au niveau KENT courant de tableau_k est nulle :

signale à l'écran qu'il s'agit d'une insertion de type N
fixe le type à N,

renvoie 1,

lorsque la longueur du dictionnaire de caractéristiques reprises
au niveau KENT courant de tableau_k, est non nulle :

signale à l'écran qu'il s'agit d'une insertion de type F
fixe le type à F,

renvoie 1.

_lorsque l'intitulé passé en argument est différent du caractère
"." :

fait rechercher parmi les rubriques de caractéristiques reprises
dans le dictionnaire de caractéristiques elles-mêmes reprises au
niveau KENT courant de tableau_k, celles dont l'intitulé est
égal ou inclut l'intitulé passé en argument :

lorsqu'aucune rubrique n'est trouvée :

affiche à l'écran, un message de non-existence,
renvoie 0,

lorsqu'une seule rubrique est trouvée :

met à jour tableau_k, des caractéristiques de la rubrique,

affiche à l'écran, l'intitulé de la rubrique,

fixe le type à E,

renvoie 1,

lorsque plusieurs rubriques sont trouvées :

en demande la sélection,

met à jour tableau_k, des caractéristiques de la rubrique

sélectionnée,

affiche à l'écran, la rubrique sélectionnée,

fixe le type à E,

renvoie 1.

(D.12) intitulé repéré :

_réinitialise chaîne_1, chaîne_2 et le nombre de rubriques trouvées dans la recherche de rubriques en cours,

_recherche parmi les rubriques de caractéristiques reprises dans le dictionnaire de caractéristiques elles-mêmes reprises au niveau BOENNINGHAUSEN courant de tableau_b lorsque le mode a pour valeur courante 1, 2 (au niveau KENT courant de tableau_k lorsque le mode a pour valeur courante 3) celles dont l'intitulé est égal ou inclut l'intitulé passé en argument,

_crée par rubrique trouvée, un élément reprenant le numéro d'ordre lui correspondant, pour l'insérer au bout de chaîne_1,

_renvoie le nombre de rubriques trouvées.

(D.13) ajouter :

_crée un élément reprenant le numéro d'ordre passé en argument pour l'insérer au bout de chaîne_1.

(D.14) charger dic :

_charge dans le buffer du fichier dictionnaire(s) concerné, l'ensemble formé des (i(passé en argument)*nombre maximum de records autorisé par le buffer)ème record et suivants autorisés par le buffer, records repris dans la "partie rubrique" du dictionnaire de niveau, adresse au sein du fichier dictionnaire(s) l'enfermant et nombre de rubriques passés en argument, le fichier dictionnaire étant celui lié au répertoire de BOENNINGHAUSEN lorsque le mode a pour valeur courante 1, 2 (au répertoire de KENT lorsque le mode a pour valeur courante 3).

(D.15) charger clé :

_charge dans un buffer prévu à cet effet, la "partie clé" de niveau, adresse au sein du fichier dictionnaire(s) l'enfermant

et longueur passées en argument, le fichier dictionnaire(s) étant celui lié au répertoire de BOENNINGHAUSEN lorsque le mode a pour valeur courante 1, 2 (au répertoire de KENT lorsque le mode a pour valeur courante 3).

(D.16) mise à neuf :

_réinitialise chaîne_1 : la libère de tous les éléments autres que l'élément de tête,

_réinitialise chaîne_2 : la libère de tous les éléments autres que l'élément de tête.

(D.17) mise à jourb :

_se positionne sur les caractéristiques de la rubrique trouvée(sélectionnée), au sein du dictionnaire les renfermant, rubrique de numéro d'ordre établi au départ des deux valeurs passées en argument :

lorsque les deux valeurs sont nulles, le numéro d'ordre se trouve repris dans l'élément directement suivant l'élément de tête de chaîne_1,

lorsque les deux valeurs sont non nulles, le numéro d'ordre se trouve repris dans l'élément de chaîne_1, (deuxième valeur-1)ème suivant de l'élément pointé par le (première valeur-1)ème élément de chaîne_2,

_reporte dans tableau_b,

au niveau BOENNINGHAUSEN courant, la caractéristique : adresse dans le fichier de la version encodée du répertoire de BOENNINGHAUSEN, de la rubrique trouvée(sélectionnée),

au niveau BOENNINGHAUSEN courant + 1, les caractéristiques : adresse dans le fichier dictionnaires de niveau correspondant, longueur et nombre de rubriques, du sous-dictionnaire attaché à la rubrique trouvée(sélectionnée),

_réinitialise tableau_b, aux niveaux supérieurs.

(D.18) mise à jourk :

_se positionne sur les caractéristiques de la rubrique trouvée(sélectionnée), au sein du dictionnaire les renfermant, rubrique de numéro d'ordre établi au départ des deux valeurs passées en argument :

lorsque les deux valeurs sont nulles, le numéro d'ordre se trouve repris dans l'élément directement suivant l'élément de tête de chaîne_1,

lorsque les deux valeurs sont non nulles, le numéro d'ordre se trouve repris dans l'élément de chaîne_1, (deuxième valeur-1)ème suivant de l'élément pointé par le (première valeur-

1)ème élément de chaîne_2,
 _reporte dans tableau_k,
 au niveau KENT courant, la caractéristique : adresse dans le
 fichier des rubriques, de la rubrique trouvée(sélectionnée),
 au niveau KENT courant + 1, les caractéristiques : adresse dans
 le fichier dictionnaires de niveau correspondant, longueur et
 nombre de rubriques, du sous-dictionnaire attaché à la rubrique
 trouvée(sélectionnée),
 _réinitialise tableau_k, aux niveaux supérieurs.

(D.19) sélectionner :

_établit au départ de chaîne_1, du nombre passé en argument de
 rubriques trouvées dans la recherche de rubriques en cours et du
 nombre maximum autorisé d'intitulés affichés dans le cas d'un
 affichage multiple : chaîne_2,
 _affiche à l'écran de façon interactive, l'intitulé des
 rubriques trouvées jusqu'à sélection de l'un d'entre eux,
 _renvoie par le biais de deux valeurs, le numéro d'ordre de la
 rubrique sélectionnée (traduisez : le numéro d'ordre se trouve
 repris dans l'élément de chaîne_1, (deuxième valeur-1)ème
 suivant de l'élément pointé par le (première valeur-1)ème
 élément de chaîne_2).

(E) les procédures de mise en forme et report :

(E.1) traiter groupe :

_réinitialise la chaîne liée aux abréviations relevées dans le
 groupe à mettre en forme,
 _retient la valeur atteinte par l'indice attaché au fichier des
 groupes(listes) mis en forme,
 _demande de façon interactive, s'il faut ou non prendre en
 considération l'entièreté des rubriques dérivées des rubriques
 constituant le groupe,
 _fait pointer l'indice attaché au fichier de la version encodée
 du répertoire de BOENNINGHAUSEN sur le caractère précédent le
 niveau de l'intitulé attaché à la rubrique définie première pour
 mettre en forme et reporter à la suite du fichier des
 groupes(listes) mis en forme, chaque intitulé, chaque Bloc-
 remèdes dans l'ordre de leur apparition dans le fichier de la
 version encodée du répertoire de BOENNINGHAUSEN, compte tenu de
 la prise ou non en considération de l'entièreté des rubriques
 dérivées,

_fait suivre le groupe mis en forme, du caractère de fin de groupe(liste),

_met à jour s'il y a lieu, la valeur du niveau KENT le plus élevé dans l'entièreté des groupes mis en forme.

(E.2) traiter liste :

_retient la valeur atteinte par l'indice attaché au fichier des groupes(listes) mis en forme,

_fait pointer l'indice attaché au fichier de la version encodée du répertoire de BOENNINGHAUSEN sur le premier caractère du Bloc-remèdes attaché à la rubrique du répertoire de BOENNINGHAUSEN à fusionner, pour mettre en forme et reporter dans le fichier des groupes(listes) mis en forme, la liste des remèdes y figurant,

_fait suivre la liste mise en forme du caractère de fin de groupe(liste).

(E.3) traiter intitulé :

_retient l'intitulé de niveau BOENNINGHAUSEN passé en argument et de premier caractère pointé par l'indice attaché au fichier de la version encodée du répertoire de BOENNINGHAUSEN pour le mettre en forme par ajout des caractères annonçant et terminant tout intitulé de la première version encodée du répertoire de KENT, établissement du niveau KENT qu'aura l'intitulé dans la première version encodée du répertoire de KENT, modification du niveau KENT établi, substitution d'abréviations et scission,

_reporte l'intitulé mis en forme à la suite du fichier des groupes(listes) mis en forme,

_retient la valeur du dernier niveau KENT établi(accepté),

_met à jour s'il y a lieu, la valeur du niveau KENT le plus haut dans l'entièreté des groupes(listes) mis en forme,

_renvoie une valeur strictement positive lorsque l'intitulé mis en forme est l'intitulé de la rubrique définie dernière du groupe,

renvoie une valeur inférieure à zéro lorsque l'intitulé mis en forme n'est pas celui de la rubrique définie dernière du groupe.

(E.4) niveau :

avec le niveau BOENNINGHAUSEN de l'intitulé retenu passé en argument

_offre lorsque s'est produit au niveau directement inférieur soit une modification de niveaux soit une scission, la possibilité de modifier le niveau KENT établi pour l'intitulé retenu pour autant que la valeur proposée ne soit ni inférieure

à celle établie ni supérieure de plus d'une unité au dernier niveau KENT établi(accepté),

_retient la valeur du dernier niveau KENT établi(accepté),

_met à jour s'il y lieu, la valeur du niveau KENT le plus haut dans l'entièreté des groupes(listes) mis en forme.

(E.5) abréviation :

substitue à toute abréviation relevée dans l'intitulé retenu (à l'exception de l'abréviation etc.) , sa correspondance :

_lorsque l'abréviation se trouve reprise dans un élément de la chaîne liée aux abréviations du groupe à mettre en forme, en propose de façon interactive, la correspondance :

si elle est acceptée : substitue à l'abréviation sa correspondance dans l'intitulé retenu,

si elle est refusée : en demande la nouvelle correspondance, substitue à l'ancienne correspondance, la nouvelle dans l'élément de la chaîne liée aux abréviations du groupe à mettre en forme, substitue à l'abréviation sa correspondance dans l'intitulé retenu.

_lorsque l'abréviation ne se trouve pas reprise dans un élément de la chaîne liée aux abréviations du groupe à mettre en forme : en demande la correspondance, crée un élément reprenant l'abréviation et sa correspondance pour l'insérer au bout de la chaîne liée aux abréviations du groupe à mettre en forme, substitue à l'abréviation sa correspondance dans l'intitulé retenu.

(E.6) scission :

scinde s'il y lieu, l'intitulé retenu à tout changement d'écriture :

_résorbe le changement d'écriture,

_propose de façon interactive, la scission :

si elle est acceptée :

scinde l'intitulé retenu pour placer de part et d'autre de la scission, les deux caractères terminant et annonçant tout intitulé de la première version encodée du répertoire de KENT, caractère lui-même suivi de la valeur du dernier niveau KENT établi(accepté) incrémenté de un,

retient la valeur du dernier niveau KENT ainsi établi,

met à jour s'il y a lieu, la valeur du niveau KENT le plus haut dans l'entièreté des groupes mis en forme.

(E.7) traiter med :

_lorsque la caractère passé en argument est "a" :

met en forme le Bloc-remèdes de premier caractère pointé par

l'indice attaché au fichier de la version encodée du répertoire de BOENNINGHAUSEN, par ajout des caractères annonçant et terminant tout Bloc-remèdes de la première version encodée du répertoire de KENT, substitution aux remèdes de la correspondance codée dans la première version encodée du répertoire de KENT, conversion des degrés /* et ajout des références */,

reporte le Bloc-remèdes mis en forme à la suite du fichier des groupes(listes) mis en forme,

_lorsque le caractère passé en argument est "k" :

met en forme la liste des remèdes de premier caractère pointé par l'indice attaché au fichier de la version encodée du répertoire de BOENNINGHAUSEN, par substitution aux remèdes de leur correspondance dans la version originale du répertoire de KENT, conversion des degrés attachés /* et ajout des références */,

reporte la liste mise en forme à la suite du fichier des groupes(listes) mis en forme.

(E.8) correspondance :

_renvoie la correspondance KENT du degré BOENNINGHAUSEN passé en argument.

(E.9) code :

_établit la correspondance dans la version originale du répertoire de KENT, la correspondance codée dans la première version encodée du répertoire de KENT et renvoie 1 lorsque le remède passé en argument se trouve repris parmi les remèdes de la version originale du répertoire de BOENNINGHAUSEN, dans le fichier des remèdes et correspondances,

_demande la correction du remède passé en argument lorsqu'il ne se trouve pas repris parmi les remèdes de la version originale du répertoire de BOENNINGHAUSEN dans le fichier des remèdes et correspondances... ce jusqu'à ce que le remède corrigé soit formé du seul caractère nul auquel cas renvoie 0 ou repris parmi les remèdes de la version originale du répertoire de BOENNINGHAUSEN, dans le fichier des remèdes et correspondances auquel cas établit la correspondance dans la version originale du répertoire de KENT, la correspondance codée dans la première version encodée du répertoire de KENT et renvoie 1.

(E.10) sve chr :

_reporte à la suite du fichier des groupes(listes) mis en forme, le caractère passé en argument.

_incrémenté d'une unité, l'indice attaché au fichier des groupes(listes) mis en forme.

(F) les procédures d'information :

(F.1) représenter :

_établit la localisation de la rubrique d'insertion dans le fichier des rubriques,

_crée un élément reprenant les quatre points de l'information relative au groupe(à la liste) mis(e) en forme pour l'insérer au sein de la chaîne d'information, conformément à l'ordre établi aux pages 43 et 44,

(F.2) déterminer :

avec passée en argument, la valeur du niveau KENT de la rubrique dernière trouvée dans l'accès en cours au répertoire de KENT, établit la localisation de la rubrique d'insertion dans le fichier des rubriques :

_véritable lorsqu'elle se trouve reprise au sein du fichier des rubriques,

_fictive (égale à la longueur du fichier des rubriques - 2) lorsqu'elle se trouve au-delà du fichier des rubriques.

(F.3) insérer :

_crée l'élément reprenant les quatre points de l'information relative au groupe(à la liste) mis(e) en forme, pour l'insérer au sein de la chaîne d'information entre les deux éléments passés en argument.

(G) les procédures de juxtaposition :

(G.1) insérer groupes :

_ouvre le fichier des groupes(listes) mis en forme à la lecture, _initialise l'indice attaché au fichier de la première version encodée du répertoire de KENT,

_pour chaque élément de la chaîne d'information, dans l'ordre de leur apparition :

reporte à la suite du fichier de la première version encodée du répertoire de KENT mise à jour, le caractère pointé par l'indice attaché au fichier de la première version encodée du répertoire de KENT et ses suivants jusqu'aux caractères (exclus) de la rubrique d'insertion mentionnée dans l'élément,

si l'élément a trait à un groupe mis en forme :

reporte à la suite du fichier de la première version encodée du répertoire de KENT mise à jour, le groupe repris dans le fichier des groupes(listes) mis en forme, à l'adresse mentionnée

dans l'élément,

si l'élément a trait à une liste mise en forme :

libère la chaîne liée aux listes de remèdes à fusionner de tous les éléments autres que l'élément de tête,

crée pour tout remède attaché à la rubrique d'insertion, un élément reprenant sa correspondance dans la version originale du répertoire de KENT, le degré affecté et la référence attachée pour l'insérer dans la chaîne liée aux listes de remèdes à fusionner, conformément à l'ordre des remèdes dans tout Bloc-remèdes de la version originale du répertoire de KENT,

crée pour tout remède de la liste reprise dans le fichier des groupes(listes) mis en forme, à l'adresse mentionnée dans l'élément, un élément reprenant le remède, le degré affecté et la référence attachée pour l'insérer dans la chaîne liée aux listes de remèdes à fusionner, conformément à l'ordre des remèdes dans tout Bloc-remèdes de la version originale du répertoire de KENT,

reporte à la suite du fichier de la première version encodée du répertoire de KENT mise à jour, l'intitulé de la rubrique d'insertion suivi du Bloc-remèdes formé des correspondances codées dans la première version encodée du répertoire de KENT, des degrés et références des remèdes repris dans chaque élément de la chaîne liée aux listes de remèdes,

_reporte à la suite du fichier de la première version encodée du répertoire de KENT mise à jour, le caractère pointé par l'indice attaché au fichier de la première version encodée du répertoire de KENT et suivants, s'il en est.

(G.2) aller en :

_reporte à la suite du fichier de la première version encodée du répertoire de KENT mise à jour, le caractère pointé par l'indice attaché au fichier de la première version encodée du répertoire de KENT et ses suivants jusqu'aux caractères (exclus) de la rubrique de niveau et adresse dans le fichier des rubriques passés en argument.

(G.3) insérer med :

_crée un élément reprenant les correspondance dans la version originale du répertoire de KENT, degré et références attachés, passés en argument, pour l'insérer dans la chaîne liée aux listes de remèdes à fusionner conformément à l'ordre des remèdes

dans tout Bloc-remèdes de la version originale du répertoire de KENT.

(G.4) comparer :

_compare les deux correspondances dans la version originale du répertoire de KENT, passées en argument,
renvoie "=" lorsque les deux correspondances se trouvent égales,
renvoie ">" lorsque la première correspondance suit la deuxième dans tout Bloc-remèdes de la version originale du répertoire de KENT,

renvoie "<" lorsque la première correspondance précède la deuxième dans tout Bloc-remèdes de la version originale du répertoire de KENT.

(G.5) get chr :

_lorsque la valeur passée en argument est 1 :
incrémente d'une unité, l'indice attaché au fichier des remèdes et correspondances,
lit et renvoie le caractère suivant du fichier des remèdes et correspondances,

_lorsque la valeur passée en argument est 2 :
incrémente d'une unité, l'indice attaché au fichier de la version encodée du répertoire de BOENNINGHAUSEN,
lit et renvoie le caractère suivant du fichier de la version encodée du répertoire de BOENNINGHAUSEN,

_lorsque la valeur passée en argument est 3 :
incrémente d'une unité, l'indice attaché au fichier de la première version encodée du répertoire de KENT,
lit et renvoie le caractère suivant du fichier de la première version encodée du répertoire de KENT.

(H) les fonctions d'écran :

(H.1) acquérir :

avec la valeur de L_ECRAN fixée à 40, les valeurs de i, mode passées en argument :

_efface à l'écran, la 20ème ligne depuis la (3+i)ème colonne lorsque mode est non nul (depuis la (L_ECRAN+3+i)ème colonne lorsque mode est nul),

_affiche à l'écran, sur la 20ème ligne en la (3+i) colonne lorsque mode est non nul (en la (L_ECRAN+3+i)ème colonne lorsque mode est nul) le caractère introduit au clavier,

_renvoie le caractère introduit au clavier.

(H.2) aff :

avec les valeurs de niv, adresse du buffer, i0, fd, ff passées en argument :

_réinitialise le numéro courant de ligne lorsqu'il dépasse le numéro autorisé pour l'affichage d'intitulé,

_efface le bas de l'écran depuis le numéro courant de ligne,

_affiche à l'écran, sur la ligne de numéro courant, depuis la (niv - valeur du niveau KENT courant + 2)ème colonne, successivement le caractère ">", le contenu du buffer depuis l'indice i0 et le caractère ":",

les caractères du buffer d'indice compris entre fd et ff (inclus) étant affichés en reverse.

(H.3) fonction :

avec la valeur de L_ECRAN fixée à 40, les valeurs de niv, i0, i1 passés en argument :

_si i0 est différent de -1 :

efface à l'écran, la (4+niv+i0)ème ligne depuis la 1ère colonne lorsque le reste de la division entière du mode courant par 2 est non nul (depuis L_ECRANème colonne lorsque le reste de la division entière du mode courant par 2 est nul),

affiche à l'écran, sur la (4+niv+i0)ème ligne depuis la (niv+2)ème colonne lorsque le reste de la division entière du mode courant par 2 est non nul (depuis la (L_ECRAN+niv+2)ème colonne lorsque le reste de la division entière du mode courant par 2 est nul), successivement le caractère ">" et l'intitulé repris (i0+1)ème dans le buffer symptome.

_efface à l'écran, la (4+niv+i1)ème ligne depuis la 1ère colonne lorsque le reste de la division entière du mode courant par 2 est non nul (depuis L_ECRANème colonne lorsque le reste de la division entière du mode courant par 2 est nul),

affiche en reverse à l'écran, sur la $(4+niv+i1)$ ème ligne depuis la $(niv+2)$ ème colonne lorsque le reste de la division entière du mode courant par 2 est non nul (depuis la $(L_ECRAN+niv+2)$ ème colonne lorsque le reste de la division entière du mode courant par 2 est nul), successivement le caractère ">" et l'intitulé repris $(i1+1)$ ème dans le buffer symptome.

(H.4) visionner niv :

avec la valeur de niv passée en argument :

_affiche à l'écran, sur la 22ème ligne en la 62ème colonne, niv.

(H.5) visionner blanc :

avec la valeur de niv passée en argument :

_efface à l'écran, les lignes de numéro compris entre $(4+niv)$ et 20 (inclus) depuis la colonne 1.

(H.6) visionner intit :

avec la valeur de L_ECRAN fixée à 40, les valeurs de cas, niv et tuple passées en argument :

_lorsque cas a pour valeur "0" :

efface à l'écran, les lignes de numéro compris entre $(4+niv)$ et 20 (inclus) depuis la 1ère colonne lorsque le reste de la division entière du mode courant par 2 est non nul (depuis la L_ECRANème colonne lorsque le reste de la division entière du mode courant par 2 est nul),

affiche à l'écran, sur la $(4+niv)$ ème ligne, depuis la $(niv+1)$ ème colonne lorsque le reste de la division entière du mode courant par 2 est non nul (depuis la $(L_ECRAN+niv+1)$ ème colonne lorsque le reste de la division entière du mode courant par 2 est nul), successivement le caractère ">" et l'intitulé de rubrique lu dans le fichier de la version encodée du répertoire de BOENNINGHAUSEN à l'adresse reprise au niveau niv de tableau_b lorsque la division du mode courant par 3 est nulle (lu dans le fichier des rubriques à l'adresse reprise au niveau niv de tableau_k lorsque la division du mode courant par 3 est on nulle).

_lorsque cas a pour valeur "Z" :

efface à l'écran, les lignes de numéro compris entre $(4+niv)$ et 20 (inclus) depuis la 1ère colonne lorsque le reste de la division entière du mode courant par 2 est non nul (depuis la L_ECRANème colonne lorsque le reste de la division entière du mode courant par 2 est nul),

affiche à l'écran, sur la $(4+niv)$ ème ligne, depuis la $(niv+1)$ ème colonne lorsque le reste de la division entière du mode courant

par 2 est non nul (depuis la $(L_ECRAN+niv+1)$ ème colonne lorsque le reste de la division entière du mode courant par 2 est nul), le message "non trouvé".

_lorsque cas a pour valeur "L" :

efface à l'écran, les lignes de numéros compris entre $(4+niv)$ et 20 (inclus) depuis la 1ère colonne lorsque le reste de la division entière de mode courant par 2 est non nul (depuis la L_ECRAN ème colonne lorsque le reste de la division entière de mode courant par 2 est nul),

affiche à l'écran, sur la $(4+niv)$ ème ligne, depuis la $(niv+1)$ ème colonne lorsque le reste de la division entière de mode courant par 2 est non nul (depuis la $(L_ECRAN+niv+1)$ ème colonne lorsque le reste de la division entière de mode courant par 2 est nul), successivement le message ">à la suite de " et l'intitulé lu dans le fichier des rubriques, de la rubrique dernière par ordre textuel parmi les rubriques de caractéristiques reprises dans le dictionnaire, de caractéristiques elles-mêmes reprises au niveau niv de tableau_k.

_lorsque cas a pour valeur "N" :

efface à l'écran, les lignes de numéro compris entre $(4+niv)$ et 20 (inclus) depuis la 1ère colonne lorsque le reste de la division entière du mode courant par 2 est non nul (depuis la L_ECRAN ème colonne lorsque le reste de la division du mode courant par 2 est nul),

affiche à l'écran, sur la $(4+niv)$ ème ligne, depuis la $(niv+1)$ ème colonne lorsque le reste de la division entière du mode courant par 2 est non nul (depuis la $(L_ECRAN+niv+1)$ ème colonne lorsque le reste de la division du mode courant par 2 est nul), successivement le message ">dérivé de " et l'intitulé de rubrique lu dans le fichier des rubriques à l'adresse reprise au niveau niv de tableau_k.

_lorsque cas a pour valeur "S" :

efface à l'écran, les lignes de numéro compris entre $(4+niv)$ et 20 (inclus) depuis la 1ère colonne lorsque le reste de la division entière du mode courant par 2 est non nul (depuis la L_ECRAN ème colonne lorsque le reste de la division entière du mode courant par 2 est nul),

affiche à l'écran sur la $(5+niv)$ ème ligne depuis la $(niv+2)$ ème colonne lorsque le reste de la division entière du mode courant par 2 est non nul (depuis la $(L_ECRAN+niv+2)$ ème colonne lorsque le reste de la division entière du mode courant par 2 est nul),

successivement le caractère ">" et l'intitulé lu dans le fichier de la version encodée du répertoire de BOENNINGHAUSEN, de la rubrique de numéro d'ordre repris dans l'élément de chaîne_1 pointé par le (uple)ème élément de chaîne_2, numéro d'ordre au sein du dictionnaire de caractéristiques reprises au niveau niv de tableau_b lorsque la division du mode courant par 3 est nulle (l'intitulé lu dans le fichier des rubriques de la rubrique de numéro d'ordre repris dans l'élément de chaîne_1 pointé par le (uple)ème élément de chaîne_2, numéro d'ordre au sein du dictionnaire de caractéristiques reprises au niveau niv de tableau_k lorsque la division du mode courant par 3 est non nulle),

affiche à l'écran, sur chaque ligne directement suivante, depuis la (niv+2)ème colonne lorsque le reste de la division entière du mode courant par 2 est non nul, (depuis la (L_ECRAN+niv+2)ème colonne lorsque le reste de la division entière du mode courant par 2 est nul), l'intitulé lu dans le fichier de la version encodée du répertoire de BOENNINGHAUSEN lorsque la division du mode courant par 3 est nulle (lu dans le fichier des rubriques lorsque la division du mode courant par 3 est non nulle) de chaque rubrique directement suivante jusqu'à concurrence du nombre maximum autorisé dans le cas d'affichage multiple,

retient dans le buffer symptome, à la suite les uns des autres, chaque intitulé affiché.

(H.7) ecran1 :

avec la valeur de cas passée en argument :

_lorsque cas a pour valeur 1 :

affiche à l'écran, l'écran figuré ci-dessous (en surimpression par opposition aux numéros de lignes, colonnes figurés normalement).

1 2

1 Mise à jour du répertoire de KENT

_lorsque cas a pour valeur 2 :
affiche à l'écran, l'écran figuré ci-dessous (en surimpression
par opposition aux numéros de lignes, colonnes figurés
normalement).

1 2

1 Mise à jour du répertoire de KENT

22

23 options : i(mmédiate) r(eportée)

(H.8) ecran2 :

avec les valeurs de cas, niv, uple passées en argument :

_lorsque cas a pour valeur 1 :

affiche à l'écran, le niveau niv avec en toile de fond la partie
d'écran figurée ci-dessous (en surimpression par opposition aux
numéros de lignes, colonnes figurés normalement).

2

21

22 mode niveau _options : < > _valeur :
23 mode intitulé _options : return /quitter

_lorsque cas a pour valeur 2 :

affiche à l'écran, le niveau niv et l'intitulé des rubriques de
niveau 1, 2 ... niv-1, lu dans le fichier de la version encodée
du répertoire de BOENNINGHAUSEN, à l'adresse reprise au niveau
correspondant dans tableau_b, avec en toile de fond l'écran
figuré ci-dessous (en surimpression par opposition aux numéros
de lignes, colonnes figurés normalement).

1 2

1 Mise à jour du répertoire de KENT

3 Délimitation/Localisation dans BOENNINGHAUSEN

21

22 mode niveau _options : < > _valeur :
23 mode intitulé _options : return /quitter

affiche à l'écran, l'intitulé des rubriques de niveau 1, 2 ... niv-1, lu dans le fichier de la version encodée du répertoire de BOENNINGHAUSEN, à l'adresse reprise au niveau correspondant de tableau_b, avec en toile de fond, la partie d'écran figurée ci-dessous (en surimpression par opposition aux numéros de lignes, colonnes figurés normalement).

2 L_ECRAN

4

21
22 mode intitulé _options : return /quitter

_lorsque cas a pour valeur 4 :
affiche à l'écran, le niveau niv et l'intitulé des rubriques de niveau 1, 2 ... niv-1, lu dans le fichier des rubriques à l'adresse reprise au niveau correspondant dans tableau_k (en surimpression par opposition aux numéros de lignes, colonnes figurés normalement).

1 2

1 Mise à jour du répertoire de KENT

3 Localisation dans KENT de l'insertion

21
22 mode niveau _options : < > _valeur :
23 mode intitulé _options : return /quitter

_lorsque cas a pour valeur 5 :
 affiche à l'écran, l'intitulé lu dans le fichier de la version
 encodée du répertoire de BOENNINGHAUSEN, de la rubrique de
 numéro d'ordre repris dans l'élément de chaine_1 pointé par le
 (uple)ème élément de chaine_2, numéro d'ordre au sein du
 dictionnaire de caractéristiques reprises au niveau niv de
 tableau_b lorsque la division du mode courant par 3 est nulle
 (lu dans le fichier des rubriques, de la rubrique de numéro
 d'ordre repris dans l'élément de chaine_1 pointé par le
 (uple)ème élément de chaine_2, numéro d'ordre au sein du
 dictionnaire de caractéristiques reprises au niveau niv de
 tableau_k lorsque la division du mode courant par 3 est non
 nulle) et l'intitulé des suivantes jusqu'à concurrence du
 nombre maximum autorisé en cas d'affichage multiple, avec en
 toile de fond, la partie d'écran figurée ci-dessous (en
 surimpression par opposition aux numéros de lignes, colonnes
 figurés normalement).

2

21
 22 **options : < > return**

(H.9) ecran3 :

avec les valeurs de cas, corr, abre et med passées en argument :

_lorsque cas a pour valeur 1 :

affiche à l'écran, l'écran figuré ci-dessous (en surimpression
 par opposition aux numéros de lignes, colonnes figurés
 normalement).

1 2

1 Mise à jour du répertoire de KENT

3 Traitement du groupe à insérer

5 ? prise en considération des rubriques dérivées

22
 23 **options : o(ui) n(on)**

_lorsque cas a pour valeur 2 :

affiche à l'écran, la partie d'écran figurée ci-dessous (en surimpression par opposition aux numéros de lignes, colonnes figurés normalement).

2

21

22 ? niveau : return

_lorsque cas a pour valeur 3 :

affiche à l'écran, l'abréviation abre et sa correspondance avec en toile de fond, la partie d'écran figurée ci-dessous (en surimpression par opposition aux numéros de lignes, colonnes figurés normalement).

2

21

22 ? est abréviation de : o(ui) n(on)

_lorsque cas a pour valeur 4 :

demande et retient la correspondance de l'abréviation affichée avec en toile de fond, la partie d'écran figurée ci-dessous (en surimpression par opposition aux numéros de lignes, colonnes figurés normalement).

2

21

22 ? correspondance de l'abréviation

_lorsque cas a pour valeur 5 :

affiche à l'écran, la partie d'écran figurée ci-dessous (en surimpression par opposition aux numéros de lignes, colonnes figurés normalement).

2

21

22 ? scission : o(ui) n(on)

_lorsque cas a pour valeur 6 :

demande et retient la correction du remède affiché, avec en toile de fond, la partie d'écran figurée ci-dessous (en surimpression par opposition aux numéros de lignes, colonnes figurés normalement).

2

21

22 ? correction du remède

En résumé, le programme formé des procédures ici décrites établit au départ du fichier de la version encodée du répertoire de BOENNINGHAUSEN, des fichiers dictionnaires de niveau 1, 2, 3 et plus du répertoire de BOENNINGHAUSEN, du fichier de la première version encodée du répertoire de KENT, des fichiers dictionnaires de niveau 1, 2, 3 et plus du répertoire de KENT, du fichier des rubriques du répertoire de KENT et du fichier des remèdes et correspondances, le fichier de la première version encodée du répertoire de KENT, mise à jour par insertion de groupes de rubriques tirées du fichier de la version encodée du répertoire de BOENNINGHAUSEN et fusion de paires de rubriques, l'une tirée du fichier de la version encodée du répertoire de BOENNINGHAUSEN, l'autre tirée du fichier de la première version encodée du répertoire de KENT.

Le fichier de la deuxième version encodée du répertoire de KENT, mise à jour, s'établit au départ du fichier de la première version encodée du répertoire de KENT, mise à jour, par le biais du programme creedic, implémenté par l' UIHN computer team dans le cadre de la mise au point du système informatique R.A.D.A.R.

4. CONCLUSION

Le mémoire ici présenté, m'a permis par le biais d'une mise en application de l'informatique au "monde extérieur", la découverte dans ses grandes lignes de l'homéopathie, la pratique du langage C et l'évolution dans un environnement UNIX, trois points loin d'être dénués d'intérêt.

5. LEGENDE

- ::= Annonce la définition du terme précédent le symbole
- \$ Indique l'alternative
- % % Indique la répétition possible 0 ou N fois des termes ainsi délimités
- CRT Indique le caractère de fin de ligne

6. REFERENCES

- (1) J.T. KENT, Repertory of the Homeopathic Materia Medica. Ed. Ehrhart et Karl, Chicago.
- (2) C.M. Boger, Boenninhausen's Characteristics and Repertory. Ed. Clara Louise Kent, M.D.
- (3) T.F.A. Allen, The Encyclopedia of Pure Materia Medica. B. Jain Publishers, New Delhi, 1977.
- (4) C. Hering, The Guiding Symptoms of our Materia Medica. B. Jain Publishers, New Delhi.
- (5) A. Jacques, Introduction à l'homéopathie Hahnemanienne. Faculté des thérapeutiques traditionnelles, Namur, 1983.
- (6) J.C. Grégoire, Le répertoire de BOENNINGHAUSEN.
- (7) J. Fichet, A. Jacques, P. Gardin, J. Paris, R.A.D.A.R. : un système expert à base de connaissances pour l'homéopathie.

7. TABLE DES MATIERES

1.	introduction	1
2.	spécifications	8
3.	démarche	9
3.1	étude de la structure du répertoire de BOENNINGHAUSEN	10
3.1.1	niveaux et intitulés de rubrique	10
3.1.2	bloc-remèdes	15
3.1.3	bloc-renvois	15
3.2	étude de la structure du répertoire de KENT	19
3.2.1	niveaux et intitulés de rubrique	19
3.2.2	bloc-remèdes	20
3.2.3	bloc-renvois	20
3.3	conversion	21
3.4	première version encodée du répertoire de KENT : étude et correspondance dans BOENNINGHAUSEN	23
3.4.1	étude	23
3.4.2	correspondance dans BOENNINGHAUSEN	24
3.5	deuxième version encodée du répertoire de KENT : étude et correspondance dans BOENNINGHAUSEN	25
3.5.1	étude	25
3.5.2	correspondance dans BOENNINGHAUSEN	27
3.5.3	implémentation de la correspondance	29
3.6	mise à jour	38
3.6.1	principe retenu pour la mise à jour de la première version encodée du répertoire de KENT	42
3.6.2	implémentation	45
4.	conclusion	73
5.	légende	74
6.	références	75
7.	table des matières	76
8.	annexes	77

8. ANNEXES

annex 1

1.	MIND	191
	TIME	222
	AGGRAVATION	223
	AMELIORATION	229
	CONCOMITANTS	229
	CROSS REFERENCE	230
	AGGRAVATION-CROSS REFERENCE	236
2.	SENSORINUM	236
	AGGRAVATION AND AMELIORATION	238
3.	VERTIGO	239
	TIME	241
	AGGRAVATION	242
	AMELIORATION	246
	CONCOMITANTS	247
	AGGRAVATION-CROSS REFERENCE	250
	CONCOMITANTS-CROSS REFERENCE	250
4.	HEAD	250
	INTERNAL	250
	TIME	280
	AGGRAVATION	281
	AMELIORATION	292
	CROSS REFERENCE	296
	AGGRAVATION-CROSS REFERENCE	297
	AMELIORATION-CROSS REFERENCE	297
	EXTERNAL	297
	MISCELLANEOUS	297
	TIME	307
	AGGRAVATION	307
	AMELIORATION	308
	CROSS REFERENCE	309
5.	EYES	309
	CROSS REFERENCE	320
	EYE BROWS	320
	EYE BROWS-CROSS REFERENCE	321
	ORBITS	321
	ORBITS-CROSS REFERENCE	322
	EYELIDS	322
	EYELIDS-CROSS REFERENCE	330
	CANTHI	330
	TIME	334
	AGGRAVATION	334
	AMELIORATION	338
	VISION	338
	TIME	345
	AGGRAVATION	346
	AMELIORATION	348
6	EARS	348
	HEARING	358
	TIME	361
	AGGRAVATION	362
	AMELIORATION	364
7	NOSE	365
	SMELL	379
	TIME	380
	AGGRAVATION	381
	AMELIORATION	382
7'	CORYZA	382
	TIME	385
	AGGRAVATION	386
	AMELIORATION	387
	CONCOMITANTS	387
8.	FACE	390
	LIPS	404
	LOWER JAW AND MAXILLARY JOINTS	410
	CHIN	412
	TIME	413
	AGGRAVATION	414
	AMELIORATION	416
9.	TEETH	416
	GUMS	429
	TIME	433
	AGGRAVATION	433
	AMELIORATION	437
	CONCOMITANTS	439
10	MOUTH	441
	PALATE	445
	THROAT (AND GULLET)	448
	SALIVA	458
	TONGUE	462

TIME	469
AGGRAVATION	469
AMELIORATION	472
11. APPETITE	472
TIME	480
12. THIRST	480
TIME	482
13. TASTE	482
TIME	489
AGGRAVATION	489
AMELIORATION	490
14. ERUCTION	490
TIME	493
AGGRAVATION	493
AMELIORATION	495
15. WATERBRASH AND HEATBURN	495
TIME	497
AGGRAVATION	497
16. HICCOUGH	498
TIME	499
AGGRAVATION	499
AMELIORATION	499
17. NAUSEA AND VOMITING	500
TIME	506
AGGRAVATION	506
AMELIORATION	510
CONCOMITANTS	511
CROSS REFERENCE	513
AGGRAVATION-CROSS REFERENCE	513
18. STOMACH	514
18' EPIGASTRIUM	522
STOMACH EPIGASTRIUM-TIME	526
STOMACH EPIGASTRIUM-	
-AGGRAVATION	526
STOMACH EPIGASTRIUM-	
-AMELIORATION	530
STOMACH EPIGASTRIUM-	
-CONCOMITANTS	532
STOMACH EPIGASTRIUM-	
-CROSS REFERENCE	533
19. HYPOCHONDRIA	533
TIME	540
AGGRAVATION	540
AMELIORATION	542
CROSS REFERENCE	543
20. ABDOMEN	543
TIME	559
AGGRAVATION	560
AMELIORATION	565
CROSS REFERENCE	567
AGGRAVATION-CROSS REFERENCE	568
AMELIORATION-CROSS REFERENCE	569
EXTERNAL ABDOMEN	569
AGGRAVATION	571
CROSS REFERENCE	571
21. INGUINAL AND PUBIC REGION	572
AGGRAVATION	575
CROSS REFERENCE	575
MONS PUBIS	575
22. FLATULENCE	576
TIME	579
AGGRAVATION	580
AMELIORATION	581
CROSS REFERENCE	581
AGGRAVATION-CROSS REFERENCE	581
23. STOOL	582
CONCOMITANTS BEFORE STOOL	593
CONCOMITANTS DURING STOOL	595
CONCOMITANTS AFTER STOOL	599
TIME	603
AGGRAVATION AND AMELIORATION	603
CROSS REFERENCE	607
CONCOMITANTS BEFORE STOOL-	
-CROSS REFERENCE	608
CONCOMITANTS DURING STOOL-	
-CROSS REFERENCE	609
CONCOMITANTS AFTER STOOL-	
-CROSS REFERENCE	609
AGGRAVATION AND AMELIORATION-	
-CROSS REFERENCE	609

24. ANUS AND RECTUM	609
CONDITIONS	615
CROSS REFERENCE	617
25. PERINEUM	617
CONDITIONS	618
CROSS REFERENCE	618
26. PROSTATE GLAND	619
27. URINE	619
SEDIMENT	623
MICTURATION	625
BEFORE URINATION	630
AT BEGINNING OF URINATION	630
DURING URINATION	631
AT CLOSE OF URINATION	633
AFTER URINATION	633
CONDITIONS OF URINATION	635
CROSS REFERENCE	636
SEDIMENT-CROSS REFERENCE	636
MICTURATION-CROSS REFERENCE	636
DURING URINATION-	
-CROSS REFERENCE	637
AFTER URINATION-	
-CROSS REFERENCE	637
28. URINARY ORGANS	637
KIDNEYS	637
UPETERS	639
BLADDER	639
URETHRA	641
MEATUS	644
CONDITIONS	644
KIDNEYS-CROSS REERENCE	645
BLADDER-CROSS REFERENCE	645
URETHRA-CROSS REFERENCE	645
29. GENITALIA	645
MALE ORGANS	646
PENIS	648
GLANS	649
PREPUCE	651
SPERAMATIC CORD	652
TESTES	653
SCROTUM	655
FEMALE ORGANS	656
TIME	664
CONDITIONS	667
MALE ORGANS-CROSS REFERENCE	667
PENIS-CROSS REFERENCE	667
GLANS-CROSS REFERENCE	667
PREPUCE-CROSS REFERENCE	668
SPERMATIC CORD-CROSS REFERENCE	668
TESTES-CROSS REFERENCE	668
SCROTUM-CROSS REFERENCE	668
FEMALE ORGANS-CROSS REFERENCE	668
CONDITIONS-CROSS REFERENCE	669
30. SEXUAL IMPLULSE	669
CONCOMITANTS OF COITION	673
CONCOMITANTS AFTER COITION	673
CONCOMITANTS AFTER POLLUTIONS	674
CONCOMITANTS AFTER COITION-	
-CROSS REFERENCE	674
CONCOMITANTS AFTER POLLUTIONS-	
-CROSS REFERENCE	674
31. MENSTRUATION	675
CONCOMITANTS BEFORE MENSES	678
CONCOMITANTS AT START OF	
MENSES	681
CONCOMITANTS DURING MENSES	682
CONCOMITANTS AFTER MENSES	686
LEUCORRHEA	687
CONCOMITANTS TO LEUCORRHEA	689
CROSS REFERENCE	690
LEUCORRHEA-CROSS REFERENCE	690
32. RESPIRATION	690
IMPEDED BY	695
TIME	698
AGGRAVATION	699
AMELIORATION	704
CONCOMITANTS	705
33. COUGH	705
EXCITED OR AGGRAVATED BY	709
AMELIORATION	719
CONCOMITANTS	720
EXPECTORATION	727

EXPECTORATION, TASTE OF	731
EXPECTORATION, ODOR OF	733
34. LARYNX AND TRACHEA	734
AGGRAVATION	738
35. VOICE AND SPEECH	738
TIME	741
CONDITIONS OF VOICE	742
36. NECK AND EXTERNAL THROAT	743
NAPE	748
TIME	751
AGGRAVATION	751
AMELIORATION	753
37. CHEST	753
INNER	753
EXTERNAL	764
AXILLAE	767
MAMMAE	769
NIPPLES	771
HEART AND REGION OF	772
TIME	777
AGGRAVATION	778
AMELIORATION	783
38. BACK	784
SCAPULAR REGION	784
BACK PROPER-DORSAL REGION	788
LUMBAR REGION-SMALL OF THE	
BACK IN GENERAL	793
SACRUM AND COCCYX	797
SPINAL COLUMN AND VERTEBRAE	799
TIME	800
AGGRAVATION	801
AMELIORATION	804
39. UPPER EXTREMITIES	805
TIME	837
AGGRAVATION	838
AMELIORATION	841
40. LOWER EXTREMITIES	842
TIME	874
AGGRAVATION	875
AMELIORATION	880
41. SENSATIONS AND COMPLAINTS	
IN GENERAL	881
42. GLANDS	937
43. BONES	940
44. SKIN AND EXTERIOR BODY	944
AGGRAVATION	980
TIME	980
45. SLEEP	980
FALLING ASLEEP	981
POSITIONS DURING SLEEP	991
WAKING	991
46. DREAMS	997
AGGRAVATION	1002
47. FEVER	1002
PATHOLOGICAL TYPES	1002
48. BLOOD	1005
49. CIRCULATION	1006
CONGESTIONS	1008
PALPITATION	1010
TIME	1012
HEART BEAT	1013
PULSE	1014
TIME	1017
AGGRAVATION	1017
CHILL	1020
PARTIAL CHILL	1022
COLDNESS	1024
PARTIAL COLDNESS	1025
SHIVERING	1030
TIME	1031
AGGRAVATION	1033
AMELIORATION	1036
CONCOMITANTS	1036
MIND	1037
HEAD, INTERNAL	1038
EXTERNAL HEAD	1038
EYES	1038
VISION	1039
EARS	1039
NOSE	1039
CORYZA	1039
FACE	1040

LIPS	1040
TEETH	1040
MOUTH AND THROAT	1041
APPETITE	1041
THIRST	1041
TASTE	1042
ERUCTION, NAUSEA,	
QUALMISHNESS AND VOMITING	1042
STOMACH, HYPOCHONDRIA, ABDOMEN	1042
STOOL	1043
URINE	1043
RESPIRATION	1043
COUGH	1044
LARYNX	1044
CHEST	1044
BACK AND LUMBAR REGION	1044
UPPER EXTREMITIES	1044
LOWER EXTREMITIES	1045
SKIN	1046
SLEEP	1046
50. HEAT AND FEVER IN GENERAL	1047
PARTIAL HEAT	1049
TIME	1059
AGGRAVATION	1060
AMELIORATION	1062
CONCOMITANTS	1063
HEAD, INTERNAL	1065
EXTERNAL HEAD	1065
EYES	1065
EARS	1066
NOSE	1066
CORYZA	1066
FACE	1066
LIPS	1067
LOWER JAW	1067
TEETH AND GUMS	1067
MOUTH	1067
THROAT	1067
APPETITE	1068
THIRST	1068
TASTE	1069
ERUCTION AND WATERBRASH	1069
NAUSEA AND VOMITING	1069
STOMACH	1069
HYPOCHONDRIA	1070
ABDOMEN	1070
FLATUS	1070
STOOL	1070
URINE	1070
RESPIRATION	1071
COUGH	1071
LARYNX	1071
EXTERNAL THROAT AND NECK	1071
CHEST	1071
BACK	1072
UPPER EXTREMITIES	1072
LOWER EXTREMITIES	1073
SENSATIONS AND GENERALITIES	1073
GLANDS	1075
BONES	1075
SKIN	1075
SLEEP	1075
51. SWEAT	1076
PARTIAL SWEAT	1079
TIME	1083
AGGRAVATION	1084
AMELIORATION	1087
CONCOMITANTS	1088
MIND	1088
HEAD	1090
EYES	1090
EARS	1090
NOSE	1090
CORIZA	1090
FACE	1091
LIPS	1091
LOWER JAW	1091
TEETH AND GUMS	1091
MOUTH	1091
THROAT	1091
APPETITE	1092
THIRST	1092

ERUCTIONS AND WATERBRASH	1092
NAUSEA AND VOMITING	1092
STOMACH	1092
HYPOCHONDRIA	1093
ABDOMEN	1093
STOOL	1093
URINE	1093
RESPIRATION	1094
COUGH	1094
LARYNX	1094
EXTERNAL THROAT AND NECK	1094
CHEST	1094
BACK	1095
UPPER EXTREMITIES	1095
LOWER EXTREMITIES	1095
GLANDS	1097
SKIN	1097
BONES	1098
SLEEP	1098
52. COMPOUND FEVER	1098
BEGINNING WITH CHILL	1098
BEGINNING WITH SHIVERING	1101
BEGINNING WITH HEAT	1101
BEGINNING WITH SWEAT	1102
53. CONDITIONS IN GENERAL	1105
TIME	1105
54. CONDITIONS OF AGGRAVATION	
AND AMELIORATION IN GENERAL	1105
55. CONCORDANCES	1154
ACONITE	1154
AGARICUS MUSCARIUS	1154
AGNUS CASTUS	1155
ALUMINA	1155
AMBRA	1156
AMMONIUM CARBONICUM	1157
AMMONIUM MURIATICUM	1157
ANACARDIUM	1158
ANGUSTURA	1159
ANTIMONIUM CRUDUM	1159
ANTIMONIUM TARTARICUM	1160
ARGENTUM METALLICUM	1160
ARNICA	1161
ARSENICUM	1162
ASAFOETIDA	1162
ASARUM	1163
AURUM	1163
BARYTA CARBONICA	1164
BELLADONA	1165
BISMUTHUM	1166
BORAX	1166
BOVISTA	1167
BRYONIA	1167
CALADIUM	1168
CALCAREA CARBONICA	1169
CAMPORA	1169
CANNABIS	1170
CANTHARIS	1171
CAPSICUM	1171
CARBO ANIMALIS	1172
CARBO VEGETALIS	1173
CAUSTICUM	1173
CHAMOMILLA	1174
CHELIDONIUM	1175
CHINA	1175
CICUTA VIROSA	1176
CINA	1177
CLEMATIS	1177
COCCULUS	1178
COFFEA	1178
COLCHICUM	1179
COLOCYNTHIS	1179
CONIUM MACULATUM	1180
CROCUS	1181
CUPRUM	1181
CYCLAMEN	1182
DIGITALIS	1182
DROSERA	1183
DULCAMARA	1183
EUPHORBIIUM	1184
EUPHRASIA	1185
FERRUM	1185
GRAPHITES	1186

GUAIACUM	1186
HELLEBORUS	1187
HEPAR SULPHURIS CALCAREUM	1188
HYOSCYAMUS	1188
IGNATIA	1189
IODUM	1190
IPECACUANHA	1190
KALI CARBONICUM	1191
KALI NITRICUM	1192
KREOSOTUM	1192
LACHESIS	1193
LAUROCERASUS	1193
LEDUM	1194
LYCOPODIUM	1195
MAGNES ARTIFICIALIS	1196
MAGNETIS POLUS ARTICUS	1196
MAGNETIS POLUS AUSTRALIS	1197
MAGNESIA CARBONICA	1197
MAGNESIA MURIATICA	1198
MANGANUM	1198
MARUM	1199
MENYANTHES	1199
MERCURIUS	1200
MEZEREUM	1200
MOSCHUS	1201
MURIATICUM ACIDUM	1202
NATRUM CARBONICUM	1202
NATRUM MURIATICUM	1203
NITRICUM ACIDUM	1203
NUX MOSCHATA	1204
NUX VOMICA	1205
OLEANDER	1205
OPIUM	1206
PARIS QUADRIFOLIA	1207
PETROLEUM	1207
PHOSPHORUS	1208
PHOSPHORICUM ACIDUM	1209
PLATINA	1209
PLUMBUM	1209
PULSATILLA	1210
RANUNCULUS BULBOSUS	1211
RANUNCULUS SCELERATUS	1211
RHEUM	1213
RHODODENDRON	1213
RHUS TOXICODENDRON	1214
RUTA GRAVEOLENS	1214
SABADILLA	1215
SABINA	1216
SAMBUCUS	1216
SARSAPARILLA	1217
SCILLA	1217
SECALE CORNUTUM	1218
SELENIUM	1219
SENEGA	1219
SEPIA	1220
SILICEA	1220
SPIGELIA	1221
SPONGIA	1222
STANNUM	1222
STAPHYSAGRIA	1223
STRAMONIUM	1224
STRONTIUM	1224
SULPHUR	1225
SULPHURICUM ACIDUM	1226
TARAXACUM	1226
THUJA	1227
VALERIANA	1228
VETRATUM ALBUM	1228
VERBASCUM	1229
VIOLA ODORATA	1230
VIOLA TRICOLOR	1230
ZINCUM	1231

annex 2

1. MIND	191
MISCELLANEOUS	191
TIME	222
AGGRAVATION	223
AMELIORATION	229
CONCOMITANTS	229
CROSS REFERENCE	230
AGGRAVATION-CROSS REFERENCE	236
2. SENSORIUM	236
MISCELLANEOUS	236
AGGRAVATION AND AMELIORATION	238
3. VERTIGO	239
MISCELLANEOUS	239
TIME	241
AGGRAVATION	242
AMELIORATION	246
CONCOMITANTS	247
AGGRAVATION-CROSS REFERENCE	250
CONCOMITANTS-CROSS REFERENCE	250
4. HEAD, INTERNAL	250
MISCELLANEOUS	250
TIME	280
AGGRAVATION	281
AMELIORATION	292
CROSS REFERENCE	296
AGGRAVATION-CROSS REFERENCE	297
AMELIORATION-CROSS REFERENCE	297
5. HEAD, EXTERNAL	297
MISCELLANEOUS	297
TIME	307
AGGRAVATION	307
AMELIORATION	308
CROSS REFERENCE	309
6. EYES	309
MISCELLANEOUS	309
CROSS REFERENCE	320
EYE BROWS	320
EYE BROWS-CROSS REFERENCE	321
ORBITS	321
ORBITS-CROSS REFERENCE	322
EYELIDS	322
EYELIDS-CROSS REFERENCE	330
7. CANTHI	330
MISCELLANEOUS	330
TIME	334
AGGRAVATION	334
AMELIORATION	338
8. VISION	338
MISCELLANEOUS	338
TIME	345
AGGRAVATION	346
AMELIORATION	348
9. EARS	348
MISCELLANEOUS	348
HEARING	359
TIME	361
AGGRAVATION	362
AMELIORATION	364
10. NOSE	365
MISCELLANEOUS	365
SMELL	379
TIME	380
AGGRAVATION	381
AMELIORATION	382
11. CORYZA	382
MISCELLANEOUS	382
TIME	385
AGGRAVATION	386
AMELIORATION	387
CONCOMITANTS	387
12. FACE	390
MISCELLANEOUS	390
LIPS	404
LOWER JAW AND MAXILLARY JOINTS	410
CHIN	412
TIME	413
AGGRAVATION	414
AMELIORATION	416

13. TEETH	416
MISCELLANEOUS	416
GUMS	429
TIME	433
AGGRAVATION	433
AMELIORATION	437
CONCOMITANTS	439
14. MOUTH	441
MISCELLANEOUS	441
PALATE	445
THROAT (AND GULLET)	448
SALIVA	458
TONGUE	462
TIME	469
AGGRAVATION	469
AMELIORATION	472
15. APPETITE	472
MISCELLANEOUS	472
TIME	480
16. THIRST	480
MISCELLANEOUS	480
TIME	482
17. TASTE	482
MISCELLANEOUS	482
TIME	489
AGGRAVATION	489
AMELIORATION	490
18. ERUCTION	490
MISCELLANEOUS	490
TIME	493
AGGRAVATION	493
AMELIORATION	495
19. WATERBRASH AND HEATBURN	495
MISCELLANEOUS	495
TIME	497
AGGRAVATION	497
20. HICCOUGH	498
MISCELLANEOUS	498
TIME	499
AGGRAVATION	499
AMELIORATION	499
21. NAUSEA AND VOMITING	500
MISCELLANEOUS	500
TIME	506
AGGRAVATION	506
AMELIORATION	510
CONCOMITANTS	511
CROSS REFERENCE	513
AGGRAVATION-CROSS REFERENCE	513
22. STOMACH	514
MISCELLANEOUS	514
EPIGASTRIUM	522
TIME	526
AGGRAVATION	526
AMELIORATION	530
CONCOMITANTS	532
STOMACH AND EPIGASTRIUM-	
-CROSS REFERENCE	533
23. HYPOCHONDRIA	533
MISCELLANEOUS	533
TIME	540
AGGRAVATION	540
AMELIORATION	542
CROSS REFERENCE	543
24. ABDOMEN	543
MISCELLANEOUS	543
TIME	559
AGGRAVATION	560
AMELIORATION	565
CROSS REFERENCE	567
AGGRAVATION-CROSS REFERENCE	568
AMELIORATION-CROSS REFERENCE	569
25. EXTERNAL ABDOMEN	569
MISCELLANEOUS	569
AGGRAVATION	571
CROSS REFERENCE	571
26. INGUINAL AND PUBIC REGION	572
MISCELLANEOUS	572
AGGRAVATION	575
CROSS REFERENCE	575
MONS PUBIS	575

27. FLATULENCE	576
MISCELLANEOUS	576
TIME	579
AGGRAVATION	580
AMELIORATION	581
CROSS REFERENCE	581
AGGRAVATION-CROSS REFERENCE	581
28. STOOL	582
MISCELLANEOUS	582
CONCOMITANTS BEFORE STOOL	593
CONCOMITANTS DURING STOOL	595
CONCOMITANTS AFTER STOOL	599
TIME	603
AGGRAVATION AND AMELIORATION	603
CROSS REFERENCE	607
CONCOMITANTS BEFORE STOOL-	
-CROSS REFERENCE	608
CONCOMITANTS DURING STOOL-	
-CROSS REFERENCE	608
CONCOMITANTS AFTER STOOL-	
-CROSS REFERENCE	609
AGGRAVATION AND AMELIORATION-	
-CROSS REFERENCE	609
29. ANUS AND RECTUM	609
MISCELLANEOUS	609
CONDITIONS	615
CROSS REFERENCE	617
30. PERINEUM	617
MISCELLANEOUS	617
CONDITIONS	618
CROSS REFERENCE	618
31. PROSTATE GLAND	619
MISCELLANEOUS	619
32. URINE	619
MISCELLANEOUS	619
SEDIMENT	623
MICTURATION	625
BEFORE URINATION	630
AT BEGINNING OF URINATION	630
DURING URINATION	631
AT CLOSE OF URINATION	633
AFTER URINATION	633
CONDITIONS OF URINATION	635
CROSS REFERENCE	636
SEDIMENT-CROSS REFERENCE	636
MICTURATION-CROSS REFERENCE	636
DURING URINATION-	
-CROSS REFERENCE	637
AFTER URINATION-	
-CROSS REFERENCE	637
33. URINARY ORGANS	637
MISCELLANEOUS	637
KIDNEYS	637
URETERS	639
BLADDER	639
URETHRA	641
MEATUS	644
CONDITIONS	644
KIDNEYS-CROSS REFERENCE	645
BLADDER-CROSS REFERENCE	645
URETHRA-CROSS REFERENCE	645
34. GENITALIA	645
MISCELLANEOUS	645
MALE ORGANS	646
PENIS	648
GLANS	649
PREPUCE	651
SPERAMATIC CORD	652
TESTES	653
SCROTUM	655
FEMALE ORGANS	656
TIME	664
CONDITIONS	667
MALE ORGANS-CROSS REFERENCE	667
PENIS-CROSS REFERENCE	667
GLANS-CROSS REFERENCE	667
PREPUCE-CROSS REFERENCE	668
SPERMATIC CORD-CROSS REFERENCE	668
TESTES-CROSS REFERENCE	668
SCROTUM-CROSS REFERENCE	668
FEMALE ORGANS-CROSS REFERENCE	668
CONDITIONS-CROSS REFERENCE	669

35. SEXUAL IMPLULSE	669
MISCELLANEOUS	669
CONCOMITANTS OF COITION	673
CONCOMITANTS AFTER COITION	673
CONCOMITANTS AFTER POLLUTIONS	674
CONCOMITANTS AFTER COITION-	
-CROSS REFERENCE	674
CONCOMITANTS AFTER POLLUTIONS-	
-CROSS REFERENCE	674
36. MENSTRUATION	675
MISCELLANEOUS	675
CONCOMITANTS BEFORE MENSES	678
CONCOMITANTS AT START OF	
MENSES	681
CONCOMITANTS DURING MENSES	682
CONCOMITANTS AFTER MENSES	686
LEUCORRHEA	687
CONCOMITANTS TO LEUCORRHEA	689
CROSS REFERENCE	690
LEUCORRHEA-CROSS REFERENCE	690
37. RESPIRATION	690
MISCELLANEOUS	690
IMPEDED BY	695
TIME	698
AGGRAVATION	699
AMELIORATION	704
CONCOMITANTS	705
38. COUGH	705
EXCITED OR AGGRAVATED BY	709
AMELIORATION	719
CONCOMITANTS	720
EXPECTORATION	727
EXPECTORATION, TASTE OF	731
EXPECTORATION, ODOR OF	733
39. LARYNX AND TRACHEA	734
MISCELLANEOUS	734
AGGRAVATION	738
40. VOICE AND SPEECH	738
MISCELLANEOUS	738
TIME	741
CONDITIONS OF VOICE	742
41. NECK AND EXTERNAL THROAT	743
MISCELLANEOUS	743
NAPE	748
TIME	751
AGGRAVATION	751
AMELIORATION	753
42. CHEST	753
INNER	753
EXTERNAL	764
AXILLAE	767
MAMMAE	769
NIPPLES	771
HEART AND REGION OF	772
TIME	777
AGGRAVATION	778
AMELIORATION	783
43. BACK	784
SCAPULAR REGION	784
BACK PROPER-DORSAL REGION	788
LUMBAR REGION-SMALL OF THE	
BACK IN GENERAL	793
SACRUM AND COCCYX	797
SPINAL COLUMN AND VERTEBRAE	799
TIME	800
AGGRAVATION	801
AMELIORATION	804
44. UPPER EXTREMITIES	805
MISCELLANEOUS	805
TIME	837
AGGRAVATION	838
AMELIORATION	841
45. LOWER EXTREMITIES	842
MISCELLANEOUS	842
TIME	874
AGGRAVATION	875
AMELIORATION	880
46. SENSATIONS AND COMPLAINTS	
IN GENERAL	881
MISCELLANEOUS	881
47. GLANDS	937
MISCELLANEOUS	937

48. BONES	940
MISCELLANEOUS	940
49. SKIN AND EXTERIOR BODY	944
MISCELLANEOUS	944
AGGRAVATION	980
TIME	980
50. SLEEP	980
FALLING ASLEEP	981
SLEEPINESS	985
CHARACTER OF	986
DURING	988
POSITIONS DURING SLEEP	991
WAKING	991
WAKING WITH	992
SLEEPLESSNESS	995
51. DREAMS	997
MISCELLANEOUS	997
AGGRAVATION	1002
52. FEVER	1002
PATHOLOGICAL TYPES	1002
53. BLOOD	1005
MISCELLANEOUS	1005
54. CIRCULATION	1006
MISCELLANEOUS	1006
CONGESTIONS	1008
PALPITATION	1010
TIME	1012
55. HEART BEAT	1013
MISCELLANEOUS	1013
PULSE	1014
TIME	1017
AGGRAVATION	1017
56. CHILL	1020
PARTIAL CHILL	1022
COLDNESS	1024
PARTIAL COLDNESS	1025
SHIVERING	1030
TIME	1031
AGGRAVATION	1033
AMELIORATION	1036
CONCOMITANTS	1036
MIND	1037
HEAD, INTERNAL	1038
EXTERNAL HEAD	1038
EYES	1038
VISION	1039
EARS	1039
NOSE	1039
CORYZA	1039
FACE	1040
LIPS	1040
TEETH	1040
MOUTH AND THROAT	1041
APPETITE	1041
THIRST	1041
TASTE	1042
ERUCTION, NAUSEA,	
QUALMISHNESS AND VOMITING	1042
STOMACH, HYPOCHONDRIA, ABDOMEN	1042
STOOL	1043
URINE	1043
RESPIRATION	1043
COUGH	1044
LARYNX	1044
CHEST	1044
BACK AND LUMBAR REGION	1044
UPPER EXTREMITIES	1044
LOWER EXTREMITIES	1045
SKIN	1046
SLEEP	1046
57. HEAT AND FEVER IN GENERAL	1047
MISCELLANEOUS	1047
PARTIAL HEAT	1049
TIME	1059
AGGRAVATION	1060
AMELIORATION	1062
CONCOMITANTS	1063
HEAD, INTERNAL	1065
EXTERNAL HEAD	1065
EYES	1065
EARS	1066
NOSE	1066

CORYZA	1066
FACE	1066
LIPS	1067
LOWER JAW	1067
TEETH AND GUMS	1067
MOUTH	1067
THROAT	1067
APPETITE	1068
THIRST	1068
TASTE	1069
ERUCTION AND WATERBRASH	1069
NAUSEA AND VOMITING	1069
STOMACH	1069
HYPOCHONDRIA	1070
ABDOMEN	1070
FLATUS	1070
STOOL	1070
URINE	1070
RESPIRATION	1071
COUGH	1071
LARYNX	1071
EXTERNAL THROAT AND NECK	1071
CHEST	1071
BACK	1072
UPPER EXTREMITIES	1072
LOWER EXTREMITIES	1073
SENSATIONS AND GENERALITIES	1073
GLANDS	1075
BONES	1075
SKIN	1075
SLEEP	1075
58. SWEAT	1076
MISCELLANEOUS	1076
PARTIAL SWEAT	1079
TIME	1083
AGGRAVATION	1084
AMELIORATION	1087
CONCOMITANTS	1088
MIND	1088
HEAD	1090
EYES	1090
EARS	1090
NOSE	1090
CORIZA	1090
FACE	1091
LIPS	1091
LOWER JAW	1091
TEETH AND GUMS	1091
MOUTH	1091
THROAT	1091
APPETITE	1092
THIRST	1092
ERUCTIONS AND WATERBRASH	1092
NAUSEA AND VOMITING	1092
STOMACH	1092
HYPOCHONDRIA	1093
ABDOMEN	1093
STOOL	1093
URINE	1093
RESPIRATION	1094
COUGH	1094
LARYNX	1094
EXTERNAL THROAT AND NECK	1094
CHEST	1094
BACK	1095
UPPER EXTREMITIES	1095
LOWER EXTREMITIES	1095
GLANDS	1097
SKIN	1097
BONES	1098
SLEEP	1098
59. COMPOUND FEVER	1098
BEGINNING WITH CHILL	1099
BEGINNING WITH SHIVERING	1101
BEGINNING WITH HEAT	1101
BEGINNING WITH SWEAT	1102
60. CONDITIONS IN GENERAL	1105
TIME	1105
61. CONDITIONS OF AGGRAVATION	
AND AMELIORATION IN GENERAL	1105
MISCELLANEOUS	1105

62. CONCORDANCES	1154
ACONITE	1154
AGARICUS MUSCARIUS	1154
AGNUS CASTUS	1155
ALUMINA	1155
AMBRA	1156
AMMONIUM CARBONICUM	1157
AMMONIUM MURIATICUM	1157
ANACARDIUM	1158
ANGUSTURA	1159
ANTIMONIUM CRUDUM	1159
ANTIMONIUM TARTARICUM	1160
ARGENTUM METALLICUM	1160
ARNICA	1161
ARSENICUM	1162
ASAFOETIDA	1162
ASARUM	1163
AURUM	1163
BARYTA CARBONICA	1164
BELLADONA	1165
BISMUTHUM	1166
BORAX	1166
BOVISTA	1167
BRYONIA	1167
CALADIUM	1168
CALCAREA CARBONICA	1169
CAMPHORA	1169
CANNABIS	1170
CANTHARIS	1171
CAPSICUM	1171
CARBO ANIMALIS	1172
CARBO VEGETALIS	1173
CAUSTICUM	1173
CHAMOMILLA	1174
CHELIDONIUM	1175
CHINA	1175
CICUTA VIROSA	1176
CINA	1177
CLEMATIS	1177
COCCULUS	1178
COFFEA	1178
COLCHICUM	1179
COLOCYNTHIS	1179
CONIUM MACULATUM	1180
CROCUS	1181
CUPRUM	1181
CYCLAMEN	1182
DIGITALIS	1182
DROSERA	1183
DULCAMARA	1183
EUPHORBIIUM	1184
EUPHRASIA	1185
FERRUM	1185
GRAPHITES	1186
GUAIAACUM	1186
HELLEBORUS	1187
HEPAR SULPHURIS CALCAREUM	1188
HYOSCYAMUS	1188
IGNATIA	1189
IODUM	1190
IPECACUANHA	1190
KALI CARBONICUM	1191
KALI NITRICUM	1192
KREOSOTUM	1192
LACHESIS	1193
LAUROCERASUS	1193
LEDUM	1194
LYCOPODIUM	1195
MAGNES ARTIFICIALIS	1196
MAGNETIS POLUS ARTICUS	1196
MAGNETIS POLUS AUSTRALIS	1197
MAGNESIA CARBONICA	1197
MAGNESIA MURIATICA	1198
MANGANUM	1198
MARUM	1199
MENYANTHES	1199
MERCURIUS	1200
MEZEREUM	1200
MOSCHUS	1201
MURIATICUM ACIDUM	1202
NATRUM CARBONICUM	1202
NATRUM MURIATICUM	1203

NITRICUM ACIDUM	1203
NUX MOSCHATA	1204
NUX VOMICA	1205
OLEANDER	1205
OPIMUM	1206
PARIS QUADRIFOLIA	1207
PETROLEUM	1207
PHOSPHORUS	1208
PHOSPHORICUM ACIDUM	1209
PLATINA	1209
PLUMBUM	1209
PULSATILLA	1210
RANUNCULUS BULBOSUS	1211
RANUNCULUS SCELERATUS	1211
RHEUM	1213
RHODODENDRON	1213
RHUS TOXICODENDRON	1214
Ruta graveolens	1214
SABADILLA	1215
SABINA	1216
SAMBUCUS	1216
SARSAPARILLA	1217
SCILLA	1217
SECALE CORNUTUM	1218
SELENIUM	1219
SENEGA	1219
SEPIA	1220
SILICEA	1220
SPIGELIA	1221
SPONGIA	1222
STANNUM	1222
STAPHYSAGRIA	1223
STRAMONIUM	1224
STRONTIUM	1224
SULPHUR	1225
SULPHURICUM ACIDUM	1226
TARAXACUM	1226
THUJA	1227
VALERIANA	1228
VETRATUM ALBUM	1228
VERBASCUM	1229
VIOLA ODORATA	1230
VIOLA TRICOLOR	1230
ZINCUM	1231

1. Les symptômes mentaux
2. Les vertiges
3. La tête
4. L'oeil
5. La vision
6. L'oreille
7. L'ouïe
8. Le nez
9. La face
10. La bouche
11. Les dents
12. La gorge
13. L'estomac
14. L'abdomen
15. Le rectum
16. Les selles
17. L'appareil urinaire
 - Vessie
 - Reins
 - Prostate
 - Uretre
 - Urines
18. L'appareil genital
19. Le larynx et la trachée
20. La respiration
21. La toux
22. L'expectoration
23. La poitrine
24. Le dos
25. Les membres
26. Le sommeil
27. Les frissons
28. La fièvre
29. La transpiration
30. La peau
31. Les généralités.

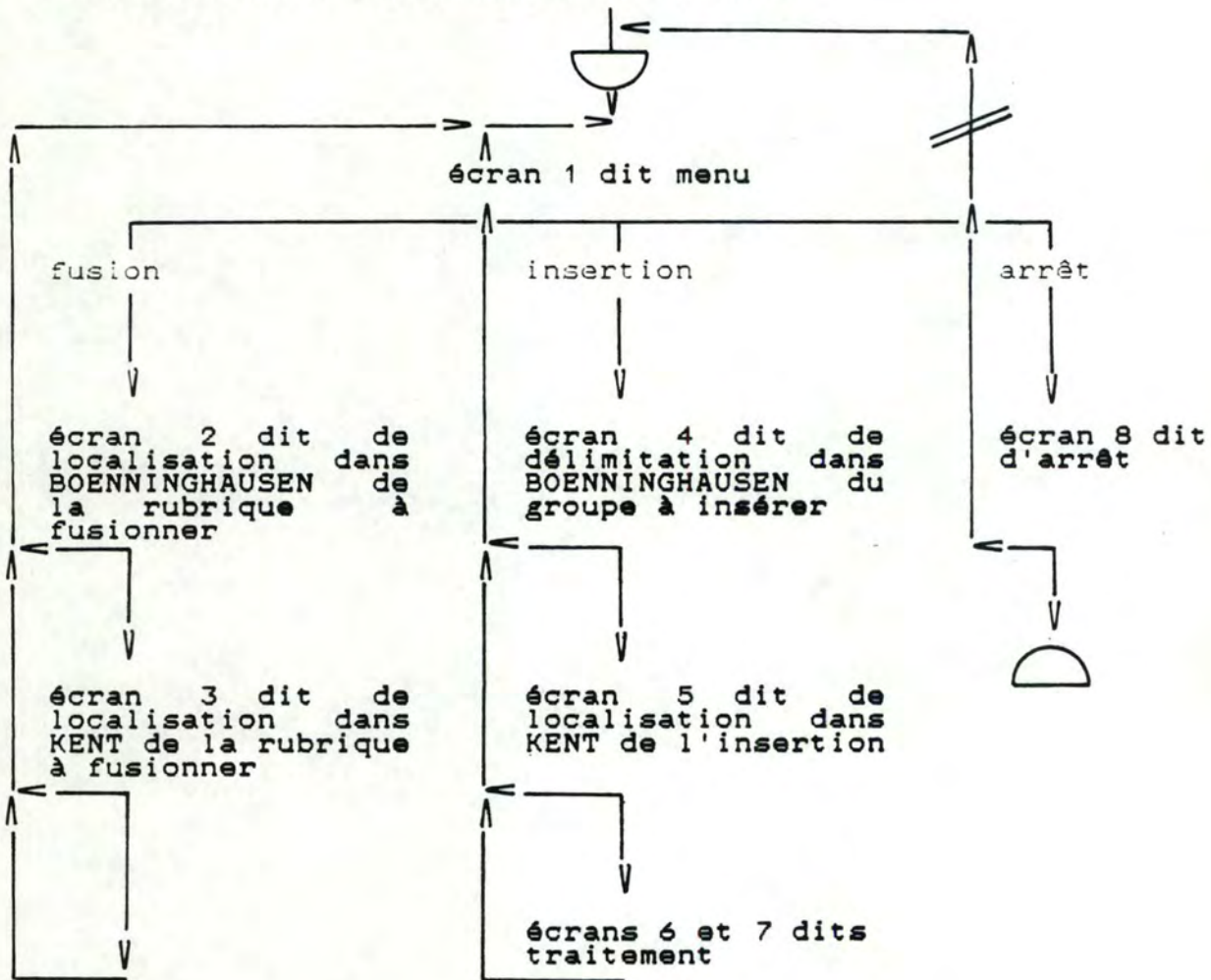
MANUEL D'UTILISATION

Le programme d'utilisation ici décrite, réalise la mise à jour du fichier de la première version encodée du répertoire de KENT par

- _fusion de paires de rubriques, entendez : insertion des remèdes attachés à une rubrique du répertoire de BOENNINGHAUSEN parmi les remèdes attachés à une rubrique du répertoire de KENT,
- _insertion de groupes de rubriques, entendez : insertion groupée de rubriques à la fois adjacentes et de même niveau au sein du répertoire de BOENNINGHAUSEN, en un endroit donné du répertoire de KENT;

le groupe de rubriques est inséré avec ou sans l'entièreté des rubriques dérivées des rubriques le constituant.

SCHEMA GENERAL D'ENCHAINEMENT DES ECRANS



Ecran 1

Mise à jour du répertoire de KENT

options : i(nsertion) --- f(usion) --- a(rrêt) -----

TAPER et 3.exe suivi de return.

APPARAÎT l'écran 1 figuré ci-contre.

VOUS SONT PROPOSÉS l'insertion d'un groupe de rubriques
la fusion de deux rubriques
l'arrêt

Ecran 2

Mise à jour du répertoire de KENT
 Délimitation/Localisation dans BOENNINGHAUSEN

```

mode niveau ----- options : < > ----- valeur : 1
mode intitulé      options : return /quitter
  
```

Ecran 2.1

Mise à jour du répertoire de KENT
 Délimitation/Localisation dans BOENNINGHAUSEN

```

>head
>external
>rheumatic pain
  
```

```

mode niveau ----- options : < > ----- valeur : 4
mode intitulé      options : return /quitter
  
```

Ecran 2.2

Mise à jour du répertoire de KENT
 Délimitation/Localisation dans BOENNINGHAUSEN

```

>head
>external
>swollen
  
```

```

mode niveau ----- options : < > ----- valeur : 4
mode intitulé      options : return /quitter
  
```

Votre choix : la fusion de deux rubriques

TAPER f non suivi de return

APPARAÎT l'écran 2 figuré ci-contre

VOUS SONT PROPOSÉS 2 modes :

*le mode niveau avec en options, les flèches gauche et droite

*le mode intitulé avec en options, return et /

Le passage d'un mode à l'autre s'effectue via les flèches haute et basse, le mode affiché en reverse étant considéré comme courant.

VOUS ÊTES DEMANDÉE via la structure même du répertoire de BOENNINGHAUSEN, la rubrique à fusionner.

Soit la rubrique d'intitulé rheumatic pain, encadré p. 305
 .veiller à ce que le mode courant soit le mode intitulé.
 .taper quelques lettres de la rubrique de niveau 1 (chapitre), en l'occurrence he pour head suivi de return.
 .taper quelques lettres de la rubrique de niveau 2, en l'occurrence ex pour external suivi de return.
 .taper quelques lettres de la rubrique de niveau 3, en l'occurrence rhe pour rheumatic pain suivi de return.
 .apparaît l'écran 2.1 figuré ci-contre.

Soit un changement d'avis de votre part quant à la rubrique à fusionner, désormais celle d'intitulé swollen, encadré page 306.

.passer en mode niveau via la flèche haute.
 .décrémenter le niveau courant via la flèche gauche, observer en parallèle la suppression de l'intitulé courant.
 .repasser en mode intitulé via la flèche basse.
 .taper quelques lettres de la rubrique de niveau 3, en l'occurrence swo pour swollen suivi de return.
 .apparaît l'écran 2.2 figuré ci-contre.

Observation 1

Le passage en mode niveau permet en cas d'erreur ou changement, de décrémenter le niveau courant jusqu'à obtention de la racine considérée comme seule valable.

Le quitter via la flèche basse.

Ecran 2.3

Mise à jour du répertoire de KENT
Délimitation/Localisation dans BOENNINGHAUSEN

- >head
 - >external
 - >swollen
 - >forehead
 - >glabella
 - >puffy
 - >sensation
 - >veins

options : <--> return-----

Ecran 2.4

Mise à jour du répertoire de KENT
Délimitation/Localisation dans BOENNINGHAUSEN

- >head
 - >external
 - >swollen
 - >f.

options : <--> return-----

Soit un nouveau changement d'avis de votre part quant à la rubrique à fusionner, désormais l'une dérivée de swollen.

.veiller à ce que le mode courant soit le mode intitulé.

.taper return.

.apparaît l'écran 2.3 figuré ci-contre.

.presser la flèche droite jusqu'à obtention de l'écran 2.4 figuré ci-contre, affichant l'intitulé de la 6ème et dernière rubrique directement dérivée de swollen. On parle encore de sous-rubrique.

.taper return lorsque l'intitulé affiché en reverse correspond à votre choix, utiliser à cet effet les flèches gauche et droite pour progresser parmi l'entière des intitulés proposés.

Observation 2

Le fait de taper un ensemble formé de 0 à N lettres, suivi de return produit :

_un affichage multiple lorsque l'ensemble s'avère débiter les intitulés de plusieurs rubriques directement dérivées de la rubrique désignée de niveau directement inférieur lorsqu'elle existe (les intitulés de plusieurs chapitres lorsqu'elle n'existe pas);

Le quitter en pressant return lorsque l'intitulé affiché en reverse correspond à votre choix.

_un affichage unique lorsque l'ensemble s'avère débiter l'intitulé d'une seule rubrique directement dérivée de la rubrique désignée de niveau directement inférieur lorsqu'elle existe (l'intitulé d'un seul chapitre lorsqu'elle n'existe pas)

_un message de non existence lorsque ou l'ensemble ne débute l'intitulé d'aucune rubrique directement dérivée de la rubrique désignée de niveau directement inférieur lorsqu'elle existe (l'intitulé d'aucun chapitre lorsqu'elle n'existe pas) ou n'existe aucune rubrique dérivée de la rubrique désignée de niveau directement inférieur.

TAPER / non suivi de return lorsque la localisation affichée à l'écran correspond à votre choix.

Ecran 3

Mise à jour du répertoire de KENT
 Localisation dans KENT de l'insertion

```

mode niveau-----options : <----->-----valeur : 1
mode intitulé      options : return    /quitter
  
```

Ecran 3.1

Mise à jour du répertoire de KENT
 Localisation dans KENT de l'insertion

```

>vertigo
>moving the head
>quickly
>amel.
  
```

```

mode niveau-----options : <----->-----valeur : 5
mode intitulé      options : return    /quitter
  
```

REAPPARAÎT l'écran 1 lorsque le symptôme mentionné ne présente pas de liste de remèdes.

APPARAÎT l'écran 3 figuré ci-contre dans le cas contraire.

VOUS SONT PROPOSÉS 2 modes :

*le mode niveau avec en options, les flèches gauche et droite

*le mode intitulé avec en options, return et /

Le passage d'un mode à l'autre s'effectue via les flèches haute et basse, le mode affiché en reverse étant considéré comme courant.

VOUS EST DEMANDÉE via la structure même du répertoire de de KENT, la rubrique à fusionner

Soit la rubrique d'intitulé amel., encadré page 101.

.veiller à ce que le mode courant soit le mode intitulé.

.taper quelques lettres de la rubrique de niveau 1 (chapitre), en l'occurrence ver pour vertigo suivi de return.

.taper quelques lettres de la rubrique de niveau 2, en l'occurrence mov pour moving the head suivi de return.

.taper quelques lettres de la rubrique de niveau 3, en l'occurrence qu pour quickly suivi de return.

.taper quelques lettres de la rubrique de niveau 4, en l'occurrence am pour amel suivi de return.

.apparaît l'écran 3.1 figuré ci-contre.

Les observations 1 et 2 sont toujours de règle.

TAPER / non suivi de return lorsque la localisation affichée à l'écran correspond à votre choix.

REAPPARAÎT l'écran 1.

Ecran 4

Mise à jour du répertoire de KENT
 Délimitation/Localisation dans BOENNINGHAUSEN

```

mode niveau ----- options : ? > ----- valeur : 1
mode intitulé ----- options : return /quitte
  
```

Ecran 4.1

Mise à jour du répertoire de KENT
 Délimitation/Localisation dans BOENNINGHAUSEN

```

>head
>aggravation
>room, in
  
```

```

mode niveau ----- options : < > ----- valeur : 4
mode intitulé ----- options : return /quitte
  
```

Ecran 4.2

Mise à jour du répertoire de KENT
 Délimitation/Localisation dans BOENNINGHAUSEN

```

>head                                     >head
>aggravation                             >aggravation
>room, in
  
```

```

mode intitulé ----- options : return ----- /quitte
  
```

Votre choix : l'insertion d'un groupe de rubriques

TAPER i non suivi de return

APPARAÎT l'écran 4 figuré ci-contre

VOUS SONT PROPOSÉS 2 modes :

*le mode niveau avec en options, les flèches gauche et droite

*le mode intitulé avec en options, return et /

Le passage d'un mode à l'autre s'effectue via les flèches haute et basse, le mode affiché en reverse étant considéré courant.

VOUS EST DEMANDÉE via la structure même du répertoire de BOENNINGHAUSEN, la 1ère rubrique du groupe

Soit à insérer le groupe formé des rubriques d'intitulés soulignés p.308

.veiller à ce que le mode courant soit le mode intitulé.

.taper quelques lettres de la rubrique de niveau 1 (chapitre), en l'occurrence he pour head suivi de return.

.taper quelques lettres de la rubrique de niveau 2, en l'occurrence ag pour aggravation suivi de return.

.taper quelques lettres de la rubrique de niveau 3, en l'occurrence ro pour room, in suivi de return.

.apparaît l'écran 4.1 figuré ci-contre.

Les observations 1 et 2 sont toujours de règle.

TAPER / non suivi de return lorsque la localisation affichée à l'écran correspond à votre choix.

APPARAÎT l'écran 4.2

VOUS EST PROPOSÉ le mode intitulé avec en options, return et /

VOUS EST DEMANDÉE la dernière rubrique du groupe

.taper quelques lettres de la rubrique de niveau 3, en l'occurrence sl pour sleep, during suivi de return.

L'observation 2 est toujours de règle.

TAPER / non suivi de return lorsque la localisation affichée à l'écran correspond à votre choix.

Ecran 5

Mise à jour du répertoire de KENT
 Localisation dans KENT de l'insertion

```

mode niveau ----- options : < --- > ----- valeur : 1
mode intitulé      options : return    /quitter
  
```

Ecran 5.1

Mise à jour du répertoire de KENT
 Localisation dans KENT de l'insertion

```

>vertigo
>looking with eyes turned
>upwards
>at high buildings
  
```

```

mode niveau ----- options : < --- > ----- valeur : 5
mode intitulé      options : return    /quitter
  
```

REAPPARAÎT l'écran 1 lorsque la rubrique choisie pour dernière précède dans le répertoire de BOENNINGHAUSEN, la rubrique choisie pour première.

APPARAÎT l'écran 5 figuré ci-contre, dans le cas contraire.

VOUS SONT PROPOSÉS 2 modes :

*le mode niveau avec en options, les flèches gauche et droite

*le mode intitulé avec en options, return et /

Le passage d'un mode à l'autre s'effectue via les flèches haute et basse, le mode affiché en reverse étant considéré courant.

VOUS EST DEMANDÉE la localisation via la structure même du répertoire de KENT, de l'insertion.

3 cas de figure se présentent à vous :

1. rendre le groupe prédéfini dérivé d'une rubrique donnée, en l'insérant immédiatement avant une sous-rubrique donnée.

Soit à rendre le groupe prédéfini dérivé de la rubrique d'intitulé upwards, encadré p. 100, en l'insérant immédiatement avant la rubrique d'intitulé at high buildings.

.veiller à ce que le mode courant soit le mode intitulé.

.taper quelques lettres de la rubrique de niveau 1 (chapitre), en l'occurrence ver pour vertigo suivi de return.

.taper quelques lettres de la rubrique de niveau 2, en l'occurrence loo pour looking with eyes turned suivi de return.

.taper quelques lettres de la rubrique de niveau 3, en l'occurrence up pour upwards suivi de return.

.taper quelques lettres de la rubrique de niveau 4, en l'occurrence at hi pour at high buildings suivi de return.

.apparaît l'écran 5.1 figuré ci-contre.

Ecran 5.2

Mise à jour du répertoire de KENT
Localisation de l'insertion

>vertigo
 >looking with eyes turned
 >upwards
 >à la suite de walking in open air, while

mode niveau	options : < >	valeur : 5
mode intitulé	options : return /quitter	

Ecran 5.3

Mise à jour du répertoire de KENT
Localisation dans KENT de l'insertion

>vertigo
 >noise, from
 >dérivé de noise, from

mode niveau	options : < >	valeur : 4
mode intitulé	options : return /	

2. rendre le groupe prédéfini dérivé d'une rubrique donnée, en l'insérant immédiatement après la dernière sous-rubrique.

Soit à rendre le groupe prédéfini dérivé de la rubrique d'intitulé upwards, encadré p.100, en l'insérant immédiatement après la dernière rubrique, en l'occurrence walking in open air, while.

.veiller à ce que le mode courant soit le mode intitulé.

.taper quelques lettres de la rubrique de niveau 1 (chapitre), en l'occurrence ver pour vertigo suivi de return.

.taper quelques lettres de la rubrique de niveau 2, en l'occurrence loo pour looking with eyes turned suivi de return.

.taper quelques lettres de la rubrique de niveau 3, en l'occurrence up pour upwards suivi de return.

.taper . suivi de return.

.apparaît l'écran 5.2 figuré ci-contre.

3. rendre le groupe prédéfini dérivé d'une rubrique donnée n'ayant encore aucune rubrique dérivée dans le répertoire non mis à jour.

Soit à rendre le groupe prédéfini dérivé de la rubrique d'intitulé noise, from encadré page 102.

.veiller à ce que le mode courant soit le mode intitulé.

.taper quelques lettres de la rubrique de niveau 1 (chapitre), en l'occurrence ver pour vertigo suivi de return.

.taper quelques lettres de la rubrique de niveau 2, en l'occurrence noi pour noise, from suivi de return.

.taper . suivi de return.

.apparaît l'écran 5.3 figuré ci-contre.

Dans les trois cas, les observations 1 et 2 sont toujours de règle.

Dans les trois cas, TAPER / non suivi de return lorsque la localisation affichée à l'écran correspond à votre choix.

Ecran 6

Mise à jour du répertoire de KENT
Traitement de l'ensemble à insérer

? prise en considération des rubriques dérivées

options : o(ni) --- n(on) -----

APPARAÎT l'écran 6 figuré ci-contre

VOUS EST DEMANDE la prise ou non en considération de l'entièreté
des rubriques dérivées des rubriques formant le groupe à insérer.

Dans l'affirmative, TAPER o non suivi de return

Dans la négative, TAPER n non suivi de return.

Ecran 7.1

Mise à jour du répertoire de KENT
 Traitement de l'ensemble à insérer

>&4room, in:
 >&4sitting:
 >&4sleep, during:

?-scission-:-(UI)-n(on)------

Ecran 7.2

Mise à jour du répertoire de KENT
 Traitement de l'ensemble à insérer

>&4room, in:
 >&4sitting:
 >&4sleep, :
 >&5during:
 >&5falling to s., when:

?-niveau-: return-----

Ecran 7.3

Mise à jour du répertoire de KENT
 Traitement de l'ensemble à insérer

>&4room, in:
 >&4sitting:
 >&4sleep:
 >&5during:
 >&6falling to s., when:

?-correspondance-de-l'abreviation-s.-:-----

APPARAÎT l'écran 7 dit de défilement parce qu'y défilent les intitulés de rubriques tels qu'ils apparaîtront dans le répertoire de KENT encodé.

VOUS EST DEMANDÉ le cas échéant

- _la correspondance d'une abréviation relevée
- _s'il y a lieu ou non de scinder un intitulé au changement d'écriture relevé
- _s'il y a lieu ou non de modifier la valeur d'un niveau établi
- _la correction d'un remède relevé non identifié

En supposant l'ensemble prédéfini page 13, l'insertion définie page 17 (cas 2), la prise en considération des rubriques dérivées :

- .apparaît l'écran 7.1 figuré ci contre
- .vous est demandé s'il y a lieu ou non de scinder
- .dans l'affirmative, taper o
- .apparaît l'écran 7.2 figuré ci-contre
- .vous est demandée la valeur du niveau
- .seront acceptés : 5 suivi de return
6 suivi de return
return (équivalent à 5 suivi de return)
- .taper 6 suivi de return
- .apparaît l'écran 7.3 figuré ci contre
- .vous est demandée la correspondance de l'abréviation s.

REAPPARAÎT l'écran 1.

Ecran 8

Mise à jour du répertoire de KENT

options : i(mmédiate) r(eportée)-----

Votre choix : l'arrêt

TAPER a non suivi de return

APPARAÎT l'écran 8 figuré ci-contre

VOUS EST DEMANDÉ s'il s'agit ou non de la dernière session.

Dans l'affirmative, TAPER i non suivi de return

Dans la négative, TAPER r non suivi de return

Observation 3

Lorsque vous est demandée une localisation au sein d'un répertoire (qu'il soit de KENT ou de BOENNINGHAUSEN), le passage immédiat en mode niveau permet via la flèche droite de récupérer les intitulés de chapitre, de sous chapitre, de rubrique... définis lors de la localisation directement précédente dans la session en cours, localisation au sein du répertoire concerné.

Observation 4

Il vous est loisible d'utiliser le programme à titre d'essai, auquel cas terminer le programme

_en choisissant l'arrêt lorsque vous est proposé l'écran 1

_en choisissant la mise à jour reportée lorsque vous est proposé l'écran 8.

Taper ensuite et2.exe suivi de return pour ainsi faire exécuter le programme supprimant l'entièreté des données que vous auriez communiquées au programme de mise à jour, depuis la dernière mise à jour.

HEAD

kali-n., Lyc., merc., mez., petr., plb., sep., sul., tab.

wobbling about: Arn., Bell., calc-p., caus., cham., chin., cup., hyo., nux-m., op., rho., spo., vio-o.

Musty odor of: Nat-m.

Nipping or pinching: Alu., (rhus-t.).

Nodes on: Ant-t., Ars., aur., Hep., kali-bi., led., mag-m., nat-m., nux-v., pho., (ran-b.), (rut.), Sil., thu.

Numb: Aco., alo., ang., asar., berb. (calc-c.), calc-p., carb-a., cast-eq., caus., chel., colo., con., daph., gel., glo., hell., kali-bro., lach., mag-m., mez., mur-ac., petr., pho., plat., sil., stic.

forehead: Sil.

side, one: Con.

occiput: Con. (protuberance), hell., kali-bro.

Pain, simple: Amb., bell., hyo., ign., kali-n., lach., mez., Nux-v., spi., stap.

Pillow feels hard: Coca.

Pimples on: Agar., am-m., ars. bar-c., bell., bov., cycl., kali-bi., kali-c., mur-ac., nat-c., old., petr., rhus-t., sil., tarx., sin.

Plica polonica: Ant-t., bar-c., bor., gras., lyc., nat-m., psor., sars., tub., ust., vino-m., vio-o., vio-t.

Pressed down, as if: Rhus-t.

Pressure: Agar., amb., anac., Arg., arn., aur., calc-c., chin., cic., clem., con., cyc., dig., dra., euphor., hyo., Lyc., mur-ac., nit-ac., Old., pho., pho-ac., rho., rhus-t., rut., saba., Sars., sep., stap., Sul., thu., sin.

Prickling: Am-c., cyc., pho-ac., vinc.

O.: Pho-ac.

EXTERNAL

Pulsation: Hell.

Pustules: Pul., sul.

Raised, as if: Hyds.

Rasping, scraping, etc.: Lyc., par.

Raw spots on: Bov., Calc-c., nit-ac., sang.

Rheumatic pain: Bar-c., stap.

Rough scalp: Rut.

Rubs the: Cam., con., hyo., tarn., ver-a.

F.: Glo., ver-a.

Scabs: Lyc., par.

Scalp, adherent: Arn., par.

Scalped, as if: Con.

Scratching, (on waking): Calc-c.

Sensitive (to touch): Aco., aesc., agar., alu., amb., am-c., ap., arg., ARS., asar., Bar-c., bell., bor., bov., Bry., calc-c., calc-p., cam., caps., carb-a., carb-v., caus., CHIN., cina., clem., cocl., cop., cup., Fer., fer-p., gel., gras., hell., Hep., ign., ip., kali-c., kali-n., kre., lach., led., lyc., mag-c., mag-m., mang., MERC., mez., nat-c., nat-m., nat-s., NIT-AC., nux-m., NUX-V., par., Petr., pho., pho-ac., plb., pul., ran-s., rho., Rhus-t., sars., scil., Sele., sep., Sil., Spi., spo., Stap., SUL., tel., thu., ver-a., vib., Zin.

V.: Chin., hep., nit-ac., nux-v., pho. Phys., scil., sil., sul., thu.

T.: Bell.

of bones: Kali-bi.

in spots: Alo., bov., nit-ac.

Separating, as if: Arg-n., bell., buf., kali-bi., lyc., mez.

Shaking scalp: Caus.

HEAD

sensation: Sil.

Compare Movements, Trembling.

Shiny scalp: Thu.

Smaller, as if: Aco., glo., grat.

Soft, as if, V. were: Petr.

Soreness, smarting V.: Ap., bov., bry.,
chel., fer., kali-bi., kali-c., kali-n.,
mag-c., mag-m., pho., ran-s., rho., sabi.,
sep., zin.

Compare Bruised Pain.

Spots, heat in: Carb-v.

Sticking: Agar., ag-c., alu., ang., ant-c.,
arn., asaf., aur., bar-c., bell., calc-c.,
canth., Caus., chin., cina, cyc., daph.,
dig., euphor., guai., hep., iod., kali-c.,
lau., mang., men., mez., Nat-m., nit-ac.,
old., par., Pho., pho-ac., ran-b., ran-s.,
rut., saba., sars., spi., stap., sul., thu.,
ver-a.

Stiff: Op., plan., rhust.

Strikes his head: Ars., hyo.:

Suppurating or festering, pain as if:
Calc-c., carb-v. (grap), iod., nit-ac.,
petr., pho., rhust., rut., stan., tarx., zin.

Sutures, pain along: Agar., calc-p., kali-bi.

Sweat on: Aco., Bell., BRY., Calc-c.,
CHAM., CHIN., cina, colo., dig.,
grap., guai., hep., Kali-c., led., MERC.,
nux-v., op., pho., PUL., rhe., rhust., sep.,
Sil., ver-a.

clommy: Cham., merc., nux-v.:

cold: Bry., cina, dig., hep., merc., ver-a.

F.: Sara.:

Swellings, tumor: Anac., arg-n., ars.,
calc-c., caus., daph., hell., kali-c., nux-v.,
petr., pho-ac., pul., rhus-t., rut.,
sep., sil.

EXTERNAL

Swollen Ap., ARS., Bell., caus., Cham.,
crot-h., cup., Daph., dig., euphor.,
glo., Grap., lach., mang., merc., mez.,
op., petr., pho., pul., RHUS-T., rut.,
sep., str., sul.

forehead: Ap., hell., ip., stan.

glabella: Flu-ac., kali-c., sele.

puffy: Ap., ars., guai., Old., sul.

sensation: Ap., berb., carb-v.

veins: Sang. (T).

F.: Abro., chin., calad., cam., cup.,
pul., sul.

Tearing: Agar., alu., amb., ang., arg.,
bar-c., bell., bry., calc-c., carb-a.,
Carb-v., colch., cyc., dig., grap., guai.,
lau., led., Lyc., men., merc., Nat-c., rho.,
rhus-t., rut., sabi., sars., sep., stap.

B.: Coc-c., Spi.

Tension of scalp: Aco., ag-c., ang., ant-t.,
Ap., arg., arn., ars., asar., bar-c., cact.,
Caus., clem., hell., kali-bro., lach.,
lau., lyc., merc., mur-ac., nit-ac., old.,
par., pho., plat., rut., saba., sep., spi.,
stan., stap., stro., tarx., Vio-o., vio-t.

Compare Contraction.

over eyes and ears: Merc.

F.: Par., pho.

V.: Par.

Thick, as if: Par.

Throbbing: Canth., guai., sul.

Thrusting pain: Ip., lach., par.

Trembling in scalp: Caus.

Compare Shaking and Jerks.

Twitching, muscular: Arg., Arn., colch.,
kali-c., lyc., mag-c.

HEAD—EXTERNAL

Headache, during: *Nit-ac.*, *rhus-t.*, *sil.*, *thu.*

after: *Plat.*

Heat, during: *Sars.*

Lying, on head: *Rhus-t.*, *thu.*

on occiput: *Nat-m.*

Mouth, opening: *Ang.*

Moving part: *Bry.*, *calc-c.*, *dig.*, *hell.*, *hyo.*,
nux-v., *spi.*, *spo.*

Noise: *Hyds.*

Overlifting: *Amb.*, *arn.*, *Calc-c.*, *Rhus-t.*

Pains, during: *Saba*

after: *Sep.*

Pressure: *Ang.*, *arg.*, *carb-v.*, *kali-n.*, *lach.*,
lyc., *mez.*

of hat: *Nit-ac.*

Rest: *Bell.*

Room, in: *Men.*

Sitting: *Dig.*, *guai.*, *pho.*, *stap.*

Sleep, during: *Cham.*, *chin.*, *rhe.*

AMELIORATION

falling to s., when: *Pul.*

Steps, ascending: *Hell.*

Stool, during: *Stap.*

Stooping: *Alu.*, *hell.*, *kali-bi.*, *mag-m.*, *stap.*

Storms, before: *Grat.*

Stormy weather: *Rho.*

Sweat, during: (*Saba*).

Talking: *Cic.*

Touch: *Amb.*, *arg.*, *arn.*, *aur.*, *bar-c.*,
Bell., *bry.*, *calc-c.*, *Chin.*, *fer.*, *grap.*,
hell., *hyo.*, *ign.*, *lyc.*, *mag-m.*, *merc.*,
nat-c., *nit-ac.*, *Nux-v.*, *par.*, *petr.*, *pho.*,
pho-ac., *rho.*, *Rhus-t.*, *sabi.*, *sars.*, *sep.*,
sil., *spi.*, *stap.*, *sul.*, *sul-ac.*, *tarx.*,
thu., *zin.*

gentle, slight, etc: *Chin.*, *LACH.*,
nux-m.

Vexation: *Mag-c.*

Weather, changing: *Bor.*

rainy: *Mag-a.*

Wind: *Bor.*, *lyc.*, *med.*, *naj.*, *Nux-v.*,
sanic.

AMELIORATION

Air, open, in: *Lau.*, *mag-m.*

Cold, externally: *Mag-m.*

Pressure: *Calc-c.*, *chin.*, *guai.*, *hell.*

hard: *Chin.*, *lach.*, *NUX-M.*

Rubbing: *Dro.*

Scratching: *Old.*

Sneezing: *Mag-m.*

Touch: *Bry.*, *dra.*

Warmth, external: *Lych.*

Wine: *Aco.*

VERTIGO

HEAT, during die: Acon., arg-m., bry., carb-v., clin., coc., croc., ign., kali-c., laur., led., mag-m., merc., mosch., nux-v., phos., puls., sep., stram., verat.
with sensation of heat in chest and about heart: Lachn.

HIGH places: Arg-n., aur., Calc., gels., nat-m., phos., puls., staph., Sulph., zinc.

HOUSE, in: Agar., am-m., arg-m., bell., croc., crot-t., lyc., mag-m., merc., mur-ac., nat-c., par., phos., puls., stann., sil., staph., sul-ac.

amel.: Agar., caust., cycl., grat., kreos., merc., sulph.

on entering: Acon., ars., carb-ac., merc., pall., phos., plat., puls., ran-b., sil., tab.
after walking: Arg-m., plat., tab.

HUNGRY, when: Dulc., kali-c.

INJURIES on head, after: Cic., nat-s.

INSPIRATION, deep, on: Cact.

INTOXICATED, as if: Acet-ac., acon., act-s., ail., agar., alum., am-c., anac., anan., arg-m., arg-n., asar., aur., bell., berb., bry., caj., camph., cann-i., carb-ac., carb-s., caust., cham., chel., chin., chin-s., cic., clem., Coc., con., cori-r., croc., crot-h., cur., dig., ferr., ferr-p., gels., glon., graph., grat., ham., hydr., hyos., kali-br., kali-c., kreos., lact., laur., led., lil-t., lyc., med., merc., merl., mez., mosch., nat-m., nux-m., Nux-v., œna., op., petr., phel., ph-ac., phos., Puls., rhod., rhus-t., sabad., sars., sec., sel., sep., sil., spig., spong., stram., tab., tarax., tep., thuj., til., valer.

KNEADING bread or making similar motions: Sanic.

KNEELING, when: Mag-c., Sep., stram.

LEANING against anything: Cycl., dig.

left cheek against hand: Verb.

head: Verb.

LEFT, swaying toward: Anac., arg-n., aur., bell., bor., calc., cic., dir., dros., eup-per., eup-pur., euph., ir-fœ., lycps., merl., mez., myris., nat-c., nux-m., sol-n., spig., sulph., zinc.

morning on waking: Myris.

evening: Nux-m.

lying, while: Merl., ox-ac.

sitting, while: Anac., merl.

standing: Merl.

LEFT, swaying toward, walking in open air: Aur., nux-m., sol-n., sulph.

LIFTING a weight: Ant-t., Puls.

LIGHTS, from being in a room with many: Nux-v.

LIGHTNING, from: Crot-h.

LOOKING with eyes turned: Spig.

downward: Alum., ars., calad., calc., camph., cham., cina, con., ferr., ferr-ar., ferr-p., graph., kalm., mag-m., merc., nat-c., nit-ac., nux-v., olnd., ox-ac., petr., Phos., puls., rhod., rhus-t., salam., sep., Spig., staph., Sulph., thuj.

as if: Phos.

either way, right or left: Con., lec., olnd., op., sabad., spig., sulph., sumb., thuj.

large pain, at a: Sep.

mirror, into a: Kali-c.

moving object, at: Agar., anac., con., cur., graph., jab., laur., mosch., nat-m., olnd., sep., sulph.

revolving objects, at: Lyc.

right, to: Lec.

steadily: All-s., am-c., ars., caust., colch., cur., kali-c., lach., manc., Nat-m., olnd., phos., sars., sil., Spig., sulph., tarent.

amel.: Dig., sabad.

straight ahead amel.: Olnd.

upwards Calc., carb-v., caust., chin-a., crot-t., cupr., dig., graph., iod., kali-p., kali-s., lach., mur-ac., nat-h., nux-v., petr., Phos., plat., plb., Puls., sang., sep., sil., stram., tab., thuj.

at high buildings: Arg-n.

at a light: Cupr., plb., thuj., zinc.

walking in open air, while: Arg-n., ox-ac., seb.

window, out of a: Camph., carb-v., Nat-m., ox-ac.

LOSS of fluids: Chin., Phos., sep.

LYING down, on (in the act of): Bell., brom., ferr., kalm., nit-ac., nux-v., olnd., ox-ac., rhus-t., sabad., sang.

necessary: Ambr., ant-t., aran., asaf., aur., chel., Coc., crot-h., cupr., graph., kali-c., kali-p., kali-s., kalm., kreos., laur., merc., mosch., nat-c., nat-m., nit-ac., op., Phos., Puls., sabin., sec., sil., sul-ac., zinc.

VERTIGO

LYING, while: Alum., am-c., *apis*, ars., aur., bar-c., brom., calad., calc., carb-v., caust., cham., coca, **Con.**, crot-c., cycl., dig., ham., iod., lac-d., lach., lact., mag-c., merc., merl., nat-s., nit-ac., nux-v., ox-ac., petr., phel., phos., pic-ac., puls., rhod., rhus-t., sang., sep., sil., spig., staph., stry., sulph., *sumb.*, *thuj.*

amel.: Acon., alum., alumn., *arn.*, aur-m., carb-an., cic., *cina*, chin., cocc., crot-h., cupr., grat., ham., kalm., lach., nat-m., nit-ac., olnd., op., petr., phel., phos., rhus-t., sil., stann., sul-ac., tell., *thuj.*

with head high amel.: Nat-m., *petr.*

as if feet were going up: Ph-ac., stict.

he did not touch the bed: Lac-c.

sinking down thro' or with the bed: Bell., benz-ac., **Bry.**, calc-p., chin-s., dulc., kali-c., lach., *lyc.*, mosch., nat-c., rhus-t., sacc.

back, on: Alum., anan., merc., merc-sul., mur-ac., nux-v., puls., sil., sulph.

amel.: Stram.

in a cool room amel.: Cast-v.

face, on, while: Phos.

amel.: Coca.

side agg.: Stram.

amel.: Merc.

right, agg.: Mur-ac.

left, agg.: Alumn., iod., phos., sil.

MEDITATING, in: Agar., arg-n., coff., gran., ph-ac., puls., sil.

amel.: Phos.

thinking of something else amel.: Agar., pip-m., sep.

walking in open air, while: Agar., sil.

MENSES, before: Acon., agn., bor., bov., bry., calc., calc-p., caul., chel., con., lach., nux-m., phos., puls., verat., zinc.

during: Acon., am-c., ant-t., arg-n., bor., bov., brom., cact., calc., calc-p., carb-s., carb-v., caul., caust., con., croc., cub., cycl., elaps, ferr., ferr-p., gels., graph., iod., kali-bi., lach., lyc., mosch., nux-v., ph-ac., plat., Puls., sec., sulph., *thuj.*, tril., uran., ust.

amel.: All-s., lach.

MENSES, during.

stooping, on: Calc., caust.

and rising again: Calc.

walking, while: Phos.

after: Agar., ant-t., con., nat-m., puls., ust.

suppressed: Acon., bry., calc., cimic., con., Cycl., gels., lach., nux-v., phos., plat., Puls., sabin., sep., sil., sulph., verat., zinc.

MENTAL exertion: Agar., agn., am-c., arg-m., arg-n., *arn.*, bar-c., bor., calc., cham., coff., cupr., gran., grat., kalm., merc-i-f., Nat-c., Nat-m., nat-p., Nux-v., ph-ac., pic-ac., puls., sep., sil., staph.

amel.: Phos.

MIRROR, after looking into: Kali-c.

MOTION, from: Agar., ail., aloe, am-c., am-m., *arn.*, aur., aur-m., bar-c., bell., **Bry.**, calc-p., carb-ac., carb-v., chin., cocc., coff., con., crot-h., crot-t., cupr., ferr-i., fl-ac., gels., glon., graph., grat., hep., hydr-ac., kalm., laur., lycps., mag-c., med., nat-a., nat-c., nat-m., nux-v., pæon., phel., phos., phys., puls., sabad., sang., sec., sel., sil., sin-n., sol-n., spong., staph., *sumb.*, tab., tell., ther.

amel.: Coff., cycl., mag-m., rhod.

arms, of: Bar-c., berb., sep.

eyes: Bell., chel., cocc., con., mur-ac., petr., plat., puls., spig.

eyelids: Alum., mosch.

sudden: Ferr., gels., lact., ptel., *sumb.*

vomiting and nausea: Selen., ther.

MOVING the head: Acon., agar., aloe, am-c., *arn.*, atro., aur., bar-c., bell., **Bry.**, calc., calc-ar., carb-an., carb-v., caust., clem., cocc., **Con.**, cupr., echi., glon., hep., ign., ip., kali-bi., kali-c., lac-d., meph., mosch., nat-m., pæon., phos., ptel., rhus-t., *samb.*, sang., sep., sel., spig., tell., ther., *thuj.*

quickly: Am-c., atro., bar-c., bry., Calc., calc-s., carb-v., coloc., gels., helo-h., kali-c., lac-ac., sang., spig., staph., sulph., verat.

amel.: Agar.

NAUSEA, with: Acon., agar., ail., alum., alumn., am-c., amyg., amyl-n., ant-c., ant-t., *apis*, arg-n., *arn.*, ars., bapt., bar-c., bell.,

NAUSEA, with.

bor., brom., bry., calad., cahin., calc., calc-p., calc-s., camph., carb-an., carb-v., caust., cham., chel., chin., chin-a., Chin-s., cemic., cinnb., coca, Cocc., coloc., con., cycl., crot-h., crot-t., Ferr., ferr-ar., ferr-p., fl-ac., gels., glon., graph., gymn., ham., hell., hep., ind., kali-ar., kali-bi., kali-br., kali-c., kali-p., kali-s., kalm., lac-c., lach., lob., lyss., lyc., mag-c., merc., mill., mosch., mur-ac., myric., nat-m., nat-s., nicc., nit-ac., nux-m., nux-v., Petr., phos., pic-ac., puls., rhus-t., rumx., sabad., sang., sanic., sars., sel., senec., sep., sil., spig., spong., squil., staph., stram., stront., sulph., tab., tarent., tell., ter., ther., verat., verat-v., vip., zinc.

morning: Calc., sabad., squil., stront.

after: Calc., cemic., gran., lyss., zinc.

closing eyes, on: Lach., ther.

looking long at one object: Sars.

lying, while: Ars.

on right side or back: Mur-ac.

with the head low: Petr.

middle of chest: Bry., phos.

motion, on: Selen.

periodic: Nat-m.

raising the head: Merc.

rising in bed, on: Bry., Cocc., verat-v.

stooping amel.: Petr.

waking, on: Spong.

NOISE, from Ther.

OBJECTS seem to be too far off: Anac., Puls., stann., stram.

seem to approach and then recede: Cic.

inverted: Bufo.

large: Caust.

move: Cocc., hydr-ac., kali-c., mosch., sep., thuj.

to the right: Lac-d., nat-s., sal-ac.

to the left and downward: Tab.

the seat on which he sat: Zinc.

reel: Anac., bell., bry., glon.

run into each other: Iris-f.

vibrate: Carb-r.

OBJECTS.

seem to turn in a circle: Agar., agn., alum., am-c., anac., arn., bar-c., bar-m., bov., bry., cadm., Chel., cic., coca, cocc., colch., con., Cycl., hell., kali-c., kali-p., kali-s., laur., lyc., mag-c., merc., merc-i-r., morph., mosch., mur-ac., Nat-m., nat-p., nat-s., olnd., nux-v., op., ph-ac., psor., rhus-t., sabad., selen., sep., sil., sol-n., sul-ac.

on looking at running water: Ferr.

room whirls: Calc., caust., cann-s., cod., dub., grat., kali-bi., Nux-m., phos., tab.

OCCIPITAL: Ang., carb-v., chin., fl-ac., Gels., glon., mcd., petr., ran-b., senec., Sil., spig., zinc.

ODOR of flowers: Hyos., Nux-v., Phos.

OLD PEOPLE, in: Ambr., bar-c., calc-p., cupr., rhus-t., sin-n.

OPENING eyes. (See Closing Eyes.)

PAINFUL: Phos., tab., tarent.

PAROXYSMAL: Agar., aloe, ant-t., arg-m., calc., caul., cupr., kali-bi., morph., nat-m., nux-v., plat., ptel., tab., sil.

PERIODICAL: Agar., ang., arg-m., camph., cocc., cycl., ign., kali-c., Nat-m., Phos., tab., ust.

every two weeks: Cocc.

PREGNANCY, during: Ars., gels., Nat-m., phos.

RAISING head: Acon., æth., ant-t., arn., bar-c., Bry., cact., calc., carb-an., carb-v., chin., clem., coloc., croc., hell., jatr., laur., mag-s., merc., merc-c., nux-v., op., phos., pic-ac., selen., stann., stram.

REACHING up, on: Bar-c., lac-d., lach., sulph.

READING, while: Am-c., ang., arg-m., arn., cupr., cur., gran., graph., grat., ham., merl., merc-i-f., par., phys., stann.

aloud: Manc., par.

too long: Arn.

walking amel.: Am-c.

after: Kali-c., ph-ac.

REELING: Acon., agar., alum., anac., ars., bell., bry., camph., caps., caust., cic., cemic., croc., cupr., ferr., glon., hydr-ac., kali-i., lach., lyc., mag-c., mag-m., nat-c., nat-m., nux-m., nux-v., ol-an., pæon., ph-ac., puls., sanic., sec., seneg., spong., stram., sulph.

/*

Directory : /usr2/demotte

Intitule : def.c

Regroupe les definitions des constantes et les declarations des structures des trois programmes et1.c, et2.c, et3.c

*/

```
#define L_ECRAN      40
#define LL_ECRAN     80
#define L_SYMPT      512
#define L_MED        20
#define L_REF        20
#define L_CLE        5000
#define L_DEB_INTIT  40
#define NIV_MAX      8
#define INTITS_PAGE  5
#define SPACE        ' '
#define _NO          "\033[0m"
#define _BO          "\033[0m\033[1m"
#define _ret         13
```

```
#define LGR_DIC1B    44
#define NBR_DIC1B    2
#define LGR_DIC1K    1088
#define NBR_DIC1K    37
```

```
#define BOE          1
#define D1_BOE       2
#define D2_BOE       3
#define D3_BOE       4
#define DN_BOE       5
#define MEDCO        7
#define KEN           8
#define D1_KEN       9
#define D2_KEN      10
#define D3_KEN      11
#define DN_KEN      12
#define RUB          13
#define KENbis       14
#define INTER        15
```

```
#define ART_D1_BOE   (sizeof(d1_boe)/DIM_D1_BOE)
#define ART_D2_BOE   (sizeof(d2_boe)/DIM_D2_BOE)
#define ART_D3_BOE   (sizeof(d3_boe)/DIM_D3_BOE)
#define ART_DN_BOE   (sizeof(dn_boe)/DIM_DN_BOE)
#define ART_D1_KEN   (sizeof(d1_ken)/DIM_D1_KEN)
#define ART_D2_KEN   (sizeof(d2_ken)/DIM_D2_KEN)
#define ART_D3_KEN   (sizeof(d3_ken)/DIM_D3_KEN)
#define ART_DN_KEN   (sizeof(dn_ken)/DIM_DN_KEN)
#define ART_INTER    (sizeof(inter)/DIM_INTER)
```

```
#define DIM_D1_BOE   DIM
#define DIM_D2_BOE   DIM
#define DIM_D3_BOE   DIM
#define DIM_DN_BOE   DIM
#define DIM_D1_KEN   DIM
#define DIM_D2_KEN   DIM
#define DIM_D3_KEN   DIM
```



```

#define DIM_DN_KEN    DIM
#define DIM_INTER     DIM
#define DIM           100

#define LGR_D1_BOE    DIM_D1_BOE*ART_D1_BOE
#define LGR_D2_BOE    DIM_D2_BOE*ART_D2_BOE
#define LGR_D3_BOE    DIM_D3_BOE*ART_D3_BOE
#define LGR_DN_BOE    DIM_DN_BOE*ART_DN_BOE
#define LGR_MEDCO     4100
#define LGR_D1_KEN    DIM_D1_KEN*ART_D1_KEN
#define LGR_D2_KEN    DIM_D2_KEN*ART_D2_KEN
#define LGR_D3_KEN    DIM_D3_KEN*ART_D3_KEN
#define LGR_DN_KEN    DIM_DN_KEN*ART_DN_KEN
#define LGR_INTER     DIM_INTER*ART_INTER

```

```

struct boel23
{
    long    adr_boe;
    long    adr_s_dic;
    long    lgr_s_dic;
    short   nbr_s_rub;
    short   n_ordre;
    short   ptr;
};

```

```

struct boen
{
    long    adr_boe;
    long    adr_s_dic;
    short   nbr_s_rub;
};

```

```

struct ken123
{
    long    adr_rub;
    long    adr_lst_med;
    long    adr_s_dic;
    short   lgr_lst_med;
    unsigned short   lgr_s_dic;
    short   nbr_s_rub;
    short   n_ordre;
    short   ptr;
};

```

```

#ifdef _VM
    short   align;
#endif

```

```

#ifdef _LM
    unsigned char   drv;
    unsigned char   vol;
#endif
};

```

```

struct kenn
{
    long    adr_rub;
    long    adr_lst_med;
    long    adr_s_dic;
    short   lgr_lst_med;
    short   lgr_s_dic;
};

```



```
};  
  
struct nemo
```

```
{  
    long    adr_dic;  
    long    lgr_dic;  
    short   nbr_rub;  
    long    adr;  
};
```

```
struct ordre
```

```
{  
    short   n_ordre;  
    struct ordre * suivant;  
};
```

```
struct ORDRE
```

```
{  
    struct ordre * p_ordre;  
    struct ORDRE  * suivant;  
};
```

```
struct liste
```

```
{  
    char    abre[20];  
    char    corr[512];  
    struct liste * suivant;  
};
```

```
struct rep
```

```
{  
    char    car;  
    long    loc;  
    struct rep * suivant;  
};
```

```
struct des
```

```
{  
    long    lock[NIV_MAX+1];  
    char    typk;  
    short   nivk;  
    long    adrg;  
    struct des * suivant;  
};
```

```
struct med
```

```
{  
    char    med[L_MED];  
    char    deg;  
    char    ref[L_REF];  
    struct med * suivant;  
};
```


井井井井井	井井井井井井井	井	井	井井井井井	井井井井井井井	井井井井井井井	井井井井井井井井
井	井	井	井井	井井	井	井	井
井	井	井井井井井井井	井	井井	井	井	井井井井井井井
井	井	井	井	井	井	井	井
井	井	井	井	井	井	井	井
井井井井井井井	井井井井井井井井井	井	井	井井井井井	井	井	井井井井井井井井

Tue May 17 11:19:58 GMT 1988

```

/*
Directory : /usr2/demotte
Intitule  : et1.c
Compile   : cc et1.c -les -o et1.exe
Fonction  : etablit au depart du fichier de la version encodee du
            repertoire de BOENNINGHAUSEN, les fichiers dictionnaires
            ce niveau 1, 2, 3 et plus, affiche a l'ecran, les longueur
            et nombre de rubriques du dictionnaire de niveau 1, etabli.
*/

```

```

#include "/usr/archimed/lib/iodef.c"
#define NBR_FLE      5
#include "/usr/archimed/lib/iodec.c"
#include "/usr2/demotte/def.c"

```

```

char * nom_bce;
char * nom_d1_boe;
char * nom_d2_boe;
char * nom_d3_boe;
char * nom_dn_boe;

```

```

char boe[ WNC_LGT];
struct boe123 d1_boe[DIM_D1_BOE];
struct boe123 d2_boe[DIM_D2_BOE];
struct boe123 d3_boe[DIM_D3_BOE];
struct boen   dn_boe[DIM_DN_BOE];

```

```

struct boe123 * p_boe123;
struct boen   * p_boen;

```

```

struct rub * deb_rub1;
struct rub * deb_rub2;
struct rub * deb_rub3;
struct rub * p_rub1;
struct rub * p_rub2;
struct rub * p_rub3;

```

```

struct cle * deb_cle1;
struct cle * deb_cle2;
struct cle * deb_cle3;
struct cle * p_cle1;
struct cle * p_cle2;
struct cle * p_cle3;

```

```

struct tab * deb_tab;
struct tab * p_tab;

```

```

struct nemo_tab nemo_tab[NIV_MAX+1];

```

```

char   cle[L_CLE];
short  lgr_cle;

```

```

short  index;
char   car;
short  niv;
short  der_niv;
short  n1;
short  n2;

```

```

short  n3;
short  n4;
short  nn;

```



```
short  n3;  
short  n4;  
short  nn;  
long   adr_boe;  
long   i_boe;  
short  i_tab;  
long   i_d2;  
long   i_d3;  
long   i_d4;  
/*
```

```

/*
main()
{
    short nbr;

    prologue();
    while(1)
    switch(car=get_chr())
    {
        case '*':while((car=get_chr())!=':');
            break;
        case '+':
        case '&':adr_boe=i_boe+2;
            nbr=(car=get_chr())-'0';
            if(nbr>niv+1) nbr=erreur1(nbr);
            switch(niv=nbr)
            {
                case 1:if(n4>0) copier_dicn();
                    if(n3>0) copier_dic3();
                    if(n2>0) copier_dic2();
                    traiter_intitule123(deb_rub1,p_rub1,deb_cle1,p_cle1,&n1);
                    break;
                case 2:if(n4>0) copier_dicn();
                    if(n3>0) copier_dic3();
                    traiter_intitule123(deb_rub2,p_rub2,deb_cle2,p_cle2,&n2);
                    break;
                case 3:if(n4>0) copier_dicn();
                    traiter_intitule123(deb_rub3,p_rub3,deb_cle3,p_cle3,&n3);
                    break;
                default:traiter_intitulen();
                    break;
            };
            while(car!=':') car=get_chr();
            break;
        case SPACE:
        case _CRT :break;
        default :erreur2();
            break;
    };
}

```

/*

```

/*
epilogue()

```



```
*/
epilogue()
{
    if(n4>0) copier_dicn();
    if(n3>0) copier_dic3();
    if(n2>0) copier_dic2();
    copier_dic1();
    _idx_inc(DN_BOE, _RCD, _EOF_MRK);
    _cls_fle(DN_BOE);
    _cls_fle(D3_BOE);
    _cls_fle(D2_BOE);
    _cls_fle(D1_BOE);
    _cls_fle(BOE);
    exit();
}
```

```
/*
```

```

*/
prologue()
{
    nom_boe = "/usr2/demotte/BOE.DAT";
    nom_d1_boe = "/usr2/demotte/D1_BOE.DAT";
    nom_d2_boe = "/usr2/demotte/D2_BOE.DAT";
    nom_d3_boe = "/usr2/demotte/D3_BOE.DAT";
    nom_dn_boe = "/usr2/demotte/DN_BOE.DAT";

    _def_file(_FILE_NAME_IS      nom_boe
              _FILE_IDENTIFIER_IS BOE
              _BUFFER_IS         boe
              _WINDOW_LENGTH_IS  _WND_LGT
              _RECORD_LENGTH_IS  1
              _ORGANISATION_IS   _SEQ
              _DENSITY_IS        _SGL
              _WHEN_EOF_USE      epilogue
              _WHEN_ERROR_USE    0
              _FOR_TRACING_USE   0);
    _opn_file(BOE _IN_MODE      _INP);

    _def_file(_FILE_NAME_IS      nom_d1_boe
              _FILE_IDENTIFIER_IS D1_BOE
              _BUFFER_IS         d1_boe
              _WINDOW_LENGTH_IS  LGR_D1_BOE
              _RECORD_LENGTH_IS  ART_D1_BOE
              _ORGANISATION_IS   _SEQ
              _DENSITY_IS        _SGL
              _WHEN_EOF_USE      0
              _WHEN_ERROR_USE    0
              _FOR_TRACING_USE   0);
    _opn_file(D1_BOE _IN_MODE   _OUT);

    _def_file(_FILE_NAME_IS      nom_d2_boe
              _FILE_IDENTIFIER_IS D2_BOE
              _BUFFER_IS         d2_boe
              _WINDOW_LENGTH_IS  LGR_D2_BOE
              _RECORD_LENGTH_IS  ART_D2_BOE
              _ORGANISATION_IS   _SEQ
              _DENSITY_IS        _SGL
              _WHEN_EOF_USE      0
              _WHEN_ERROR_USE    0
              _FOR_TRACING_USE   0);
    _opn_file(D2_BOE _IN_MODE   _OUT);

    _def_file(_FILE_NAME_IS      nom_d3_boe
              _FILE_IDENTIFIER_IS D3_BOE
              _BUFFER_IS         d3_boe
              _WINDOW_LENGTH_IS  LGR_D3_BOE
              _RECORD_LENGTH_IS  ART_D3_BOE
              _ORGANISATION_IS   _SEQ
              _DENSITY_IS        _SGL
              _WHEN_EOF_USE      0
              _WHEN_ERROR_USE    0
              _FOR_TRACING_USE   0);
    _opn_file(D3_BOE _IN_MODE   _OUT);

    _def_file(_FILE_NAME_IS      nom_dn_boe
              _FILE_IDENTIFIER_IS DN_BOE
              _BUFFER_IS         dn_boe
              _WINDOW_LENGTH_IS  LGR_DN_BOE
              _RECORD_LENGTH_IS  ART_DN_BOE
              _ORGANISATION_IS   _SEQ

```



```

        _WINDOW_LENGTH_IS      LGR_DN_BOE
        _RECORD_LENGTH_IS      ART_DN_BOE
        _ORGANISATION_IS      _SEQ
        _DENSITY_IS          _SGL
        _WHEN_EOF_USE          0
        _WHEN_ERROR_USE        0
        _FOR_TRACING_USE        0);
_opn_file(DN_BOE _IN_MODE      _OUT);

niv          =99;
der_niv      =0;
n1=n2=n3=n4=nn=0;

i_boe        = -1;
i_d2=i_d3=i_d4=0;
i_tab        =1;

deb_rub1=(struct rub *)malloc(sizeof(struct rub));
deb_rub1->suiwant=0;
p_rub1=deb_rub1;

deb_rub2=(struct rub *)malloc(sizeof(struct rub));
deb_rub2->suiwant=0;
p_rub2=deb_rub2;

deb_rub3=(struct rub *)malloc(sizeof(struct rub));
deb_rub3->suiwant=0;
p_rub3=deb_rub3;

deb_cle1=(struct cle *)malloc(sizeof(struct cle));
deb_cle1->intit[0]='\0';
deb_cle1->suiwant=0;
p_cle1=deb_cle1;

deb_cle2=(struct cle *)malloc(sizeof(struct cle));
deb_cle2->intit[0]='\0';
deb_cle2->suiwant=0;
p_cle2=deb_cle2;

deb_cle3=(struct cle *)malloc(sizeof(struct cle));
deb_cle3->intit[0]='\0';
deb_cle3->suiwant=0;
p_cle3=deb_cle3;

deb_tab=(struct tab *)malloc(sizeof(struct tab));
deb_tab->i_tab =0;
deb_tab->suiwant=0;
p_tab=deb_tab;
}

```

/*

```

*/
erreur1(nbr)
{
    short nbr;
    char * s;
    char ret;

    B_clear(_SCREEN, _ALL, 0, 0, 0);
    B_put_screen();
    s="rupture de niveaux";
    B_locate(4,1);
    B_sprintf(s);
    s="
";
    B_locate(5,1);
    B_underscore();
    B_sprintf(s);
    B_normal();
    s="? niveau Boenninghausen de l'intitule";
    B_locate(8,1);
    B_sprintf(s);
    B_put_screen();
    affichage_hierarchise();
    while(nbr>niv+1)
    {
        B_locate(4,66);
        B_put_screen();
        scanf("%d%c",&nbr,&ret);
    };
    _red_fle(BOE, _BYT, _WND_LGT, adr_boe);
    return(nbr);
}

```

```

erreur2()
{
    char * s;

    B_clear(_SCREEN, _ALL, 0, 0, 0);
    B_put_screen();
    s="symbole d'entete inconnu";
    B_locate(4,1);
    B_sprintf(s);
    s="
";
    B_locate(5,1);
    B_underscore();
    B_sprintf(s);
    B_normal();
    B_put_screen();
    affichage_hierarchise();
    exit();
}

```

```

erreur3()
{
    char * s;

    B_clear(_SCREEN, _ALL, 0, 0, 0);
    B_put_screen();
    s="intitule double";

```

```

    B_locate(4,1);
    B_sprintf(s);
    s="
";

```



```

B_locate(4,1);
B_sprintf(s);
s="
B_locate(5,1);
B_underscore();
B_sprintf(s);
B_normal();
B_put_screen();
affichage_hierarchise();
exit();
}

```

```

affichage_hierarchise()

```

```

{
short i;
struct rub * p_rub;
struct tab * ptr;

i= 0;
while(++i<=niv)
switch(i)
{
case 1: p_rub=deb_rub1;
while(p_rub->suiwant!=0) p_rub=p_rub->suiwant;
affichage(p_rub->adr_boe,1);
break;

case 2: p_rub=deb_rub2;
while(p_rub->suiwant!=0) p_rub=p_rub->suiwant;
affichage(p_rub->adr_boe,2);
break;

case 3: p_rub=deb_rub3;
while(p_rub->suiwant!=0) p_rub=p_rub->suiwant;
affichage(p_rub->adr_boe,3);
break;

default: ptr=deb_tab;
while(ptr->i_tab!=nemo_tab[i].i_tab+nemo_tab[i].n-1)
ptr=ptr->suiwant;
affichage(ptr->adr_boe,i);
break;

};
if(adr_boe!=(niv<=3 ? p_rub->adr_boe : ptr->adr_boe))
affichage(adr_boe,i);
}

```

```

affichage(adr_boe,espace)

```

```

long adr_boe;
short espace;
{
short i;

_red_fle(BOE, _BYT, _WND_LGT, adr_boe);
B_locate(9+espace,espace);
B_put_screen();
for(i=0; i<LL_ECRAN-3-espace; i++)
switch(car=_get_nxt_chr(BOE))
{
case _CRT:
case '[': break;

```

```
case '?': return;  
default : printf("%c",car);  
break;
```

```
};  
printf("...");  
}
```

```
/*
```

```
*/  
get_chr()
```



```
*/  
get_chr()  
{  
    i_boe++;  
    return(car=_get_nxt_chr(BOE));  
}  
/*
```

```

*/
copier_dic3()
{
    etablier_cle(deb_cle3);
    copier_rub(3,deb_rub3,deb_cle3);
    copier_cle(3);
    mise_a_jour(2);
    mise_a_neuf123(deb_rub3,p_rub3,deb_cle3,p_cle3,&n3);
}

copier_dic2()
{
    etablier_cle(deb_cle2);
    copier_rub(2,deb_rub2,deb_cle2);
    copier_cle(2);
    mise_a_jour(1);
    mise_a_neuf123(deb_rub2,p_rub2,deb_cle2,p_cle2,&n2);
}

copier_dic1()
{
    etablier_cle(deb_cle1);
    copier_rub(1,deb_rub1,deb_cle1);
    copier_cle(1);
    mise_a_jour(0);
}

etablier_cle(deb_cle)
{
    struct cle * deb_cle;
    {
        struct cle * p_cle_prec;
        struct cle * p_cle;
        struct cle * p_cle_suiv;

        lgr_cle=0;
        p_cle      =deb_cle;
        p_cle_suiv=deb_cle->suivant;
        while(p_cle_suiv->suivant!=0)
        {
            p_cle_prec=p_cle;
            p_cle      =p_cle_suiv;
            p_cle_suiv=p_cle->suivant;
            p_cle->adr=lgr_cle;
            cas_2(0,p_cle_prec->intit,p_cle->intit,p_cle_suiv->intit);
        };
        p_cle_prec=p_cle;
        p_cle      =p_cle_suiv;
        p_cle->adr=lgr_cle;
        cas_1(0,p_cle_prec->intit,p_cle->intit);
    }
}

cas_2(i,intit_prec,intit,intit_suiv)
{
    short i;
    char intit_prec[];
    char intit[];
    char intit_suiv[];

    {
        if((intit_prec[i]==intit[i])&&(intit[i]==intit_suiv[i]))
    }

```



```

{
if((intit_prec[i]==intit[i])&&(intit[i]==intit_suiv[i]))
{
cle[lgr_cle++]=intit[i++];
cas_2(i,intit_prec,intit,intit_suiv);
}
else
{
if((intit_prec[i]==intit[i])&&(intit[i]!=intit_suiv[i]))
{
cle[lgr_cle++]=intit[i++];
cas_1(i,intit_prec,intit);
}
else
{
if((intit_prec[i]!=intit[i])&&(intit[i]==intit_suiv[i]))
{
cle[lgr_cle++]=intit[i++];
cas_1(i,intit_suiv,intit);
}
else
if((intit_prec[i]!=intit[i])&&(intit[i]!=intit_suiv[i]))
{
cle[lgr_cle++]=intit[i++];
cle[lgr_cle++]='\0';
};
};
};
};
};

```

```

cas_1(i,intit_,intit)
short i;
char intit_[];
char intit[];
{
while(intit_[i]==intit[i])
cle[lgr_cle++]=intit[i++];
cle[lgr_cle++]=intit[i];
cle[lgr_cle++]='\0';
}

```

```

copier_rub(n,ceb_rub,deb_cle)
short n;
struct rub * deb_rub;
struct cle * deb_cle;
{
struct rub * p_rub;
struct cle * p_cle;

p_rub=deb_rub;
p_cle=deb_cle;
while(p_rub->suivant!=0)
{
p_rub=p_rub->suivant;
p_cle=p_cle->suivant;
index=_idx_inc(D1_BOE+n-1,_RCD,_OUT);
p_boe123=(struct boe123 *)
(_BUF(D1_BOE+n-1)+index*_RCD_DEF(D1_BOE+n-1));
}
}

```

```

p_boe123->adr_boe =p_rub->adr_boe;
p_boe123->adr_s_dic=p_rub->adr_s_dic;
p_boe123->lgr_s_dic=p_rub->lgr_s_dic;
p_boe123->nbr_s_rub=p_rub->nbr_s_rub;
p_boe123->n_ordre =p_cle->n_ordre;
p_boe123->ptr      =p_cle->adr;
};

```

```

}

```

```

copier_cle(n)

```

```

short n;
{
char * adr_buf;

_idx_inc(D1_BOE+n-1,_RCD,_EOF_MRK);
adr_buf=_BUF(D1_BOE+n-1);
_BUF(D1_BOE+n-1)=cle;
_wrt_fle(D1_BOE+n-1,_BYT,lgr_cle,_BYT_LOC(D1_BOE+n-1));
_BYT_LOC(D1_BOE+n-1) += lgr_cle;
_BUF(D1_BOE+n-1)=adr_buf;
}

```

```

mise_a_neuf123(deb_rub,p_rub,deb_cle,p_cle,p_n)

```

```

struct rub * deb_rub, * p_rub;
struct cle * deb_cle, * p_cle;
short * p_n;
{
p_rub=deb_rub;
p_cle=deb_cle;
while(p_rub->suivant!=0)
{
p_rub=p_rub->suivant;
p_cle=p_cle->suivant;
free(p_rub);
free(p_cle);
};

* p_n=0;
deb_rub->suivant=0;
p_rub=deb_rub;
deb_cle->suivant=0;
p_cle=deb_cle;
}

```

```

mise_a_jour(report)

```

```

short report;
{
struct rub * p_rub;

switch(report)
{
case 0: printf("\nlgr du dic de niv 1 : %d",n1*_RCD_DEF(D1_BOE)+lgr_cle);
printf("\nnbr de rub de niv 1 : %d\n",n1);
break;
case 1: p_rub=deb_rub1;
while(p_rub->suivant!=0) p_rub=p_rub->suivant;
p_rub->adr_s_dic=i_d2;
p_rub->lgr_s_dic=n2*_RCD_DEF(D2_BOE)+lgr_cle;

```

```

p_rub->nbr_s_rub=n2;
i_d2 += n2*_RCD_DEF(D2_BOE)+lgr_cle;

```



```

        p_rub->nbr_s_rub=n2;
        i_d2 += n2*_RCD_DEF(D2_BOE)+lgr_cle;
        break;
case 2: p_rub=deb_rub2;
        while(p_rub->suivant!=0) p_rub=p_rub->suivant;
        p_rub->adr_s_dic=i_d3;
        p_rub->lgr_s_dic=n3*_RCD_DEF(D3_BOE)+lgr_cle;
        p_rub->nbr_s_rub=n3;
        i_d3 += n3*_RCD_DEF(D3_BOE)+lgr_cle;
        break;
case 3: p_rub=deb_rub3;
        while(p_rub->suivant!=0) p_rub=p_rub->suivant;
        p_rub->adr_s_dic=i_d4;
        p_rub->lgr_s_dic=nn;
        p_rub->nbr_s_rub=n4;
        i_d4 += nn*_RCD_DEF(DN_BOE);
        break;
};
}

```

```

traiter_intitule123(deb_rub,p_rub,deb_cle,p_cle,p_n)

```

```

    struct rub * deb_rub, * p_rub;
    struct cle * deb_cle, * p_cle;
    short * p_n;
    {
        short i;
        char intit[L_DEB_INTIT];
        struct cle * p_cle_prec, * p_cle_mem;

        p_rub=deb_rub;
        while(p_rub->suivant!=0) p_rub=p_rub->suivant;
        p_rub->suivant=(struct rub *)malloc(sizeof(struct rub));
        p_rub=p_rub->suivant;
        p_rub->adr_boe =adr_boe;
        p_rub->adr_s_dic=0;
        p_rub->lgr_s_dic=0;
        p_rub->nbr_s_rub=0;
        p_rub->suivant =0;

```

```

        printf("\n");
        for(i=0; i<L_DEB_INTIT; i++)
            switch(car=get_chr())
            {
                case _CRT :
                case '[' : break;
                case ':' : intit[i]=': ';
                        i=L_DEB_INTIT;
                        break;
                default : if(_UPR_CAS_CHR(car)) car+=32;
                        intit[i]=car;
                        printf("%c",car);
                        break;
            };

```

```

        p_cle_prec=deb_cle;
        p_cle      =deb_cle->suivant;
        while(p_cle!=0)
        {
            if(sup(p_cle->intit,intit)) break;

```

```

        p_cle_prec=p_cle;
        p_cle=p_cle->suiwant;
    };
    p_cle_mem=p_cle_prec->suiwant;
    p_cle_prec->suiwant=(struct cle *)malloc(sizeof(struct cle));
    p_cle=p_cle_prec->suiwant;
    p_cle->n_ordre= (* p_n) ++;
    p_cle->adr      =0;
    for(i=0; i<L_DEB_INTIT; i++)
    {
        p_cle->intit[i]=intit[i];
        if(intit[i]==':') break;
    };
    p_cle->suiwant=p_cle_mem;
}

```

```

sup(intit_,intit)
char intit_[];
char intit[];
{
    short i;

    for(i=0; i<L_DEB_INTIT; i++)
    {
        if(intit_[i]==':') break;
        if(intit[i]==':') break;
        if(intit_[i]>intit[i]) return(1);
        if(intit_[i]<intit[i]) return(0);
    };
    /* duplicate key */
    erreur3();
}

```

```

/*

```

```

*/
copier_dicn()

```



```

*/
copier_dicn()
{
    copier_sup(3);
    mise_a_jour(3);
    mise_a_neufn();
}

traiter_intitulen()
{
    if(niv>der_niv)
    {
        nemo_tab[niv].i_tab=i_tab;
        nemo_tab[niv].n      =0;
        p_tab=deb_tab;
        while(p_tab->i_tab!=i_tab-1) p_tab=p_tab->suiwant;
        if(p_tab->suiwant==0)
        {
            p_tab->suiwant=(struct tab *)malloc(sizeof(struct tab));
            p_tab=p_tab->suiwant;
            p_tab->suiwant=0;
        }
        else p_tab=p_tab->suiwant;}
    else
    {
        if(niv<der_niv)
        {
            copier_sup(niv);
            i_tab=nemo_tab[niv+1].i_tab;
            p_tab=deb_tab;
            while(p_tab->i_tab!=i_tab) p_tab=p_tab->suiwant;
        }
        else
        {
            if(p_tab->suiwant==0)
            {
                p_tab->suiwant=(struct tab *)malloc(sizeof(struct tab));
                p_tab=p_tab->suiwant;
                p_tab->suiwant=0;
            }
            else p_tab=p_tab->suiwant;
        }
    };
    if(niv==4) n4++;
    nemo_tab[niv].n++;
    der_niv=niv;
    p_tab->adr_boe  =adr_boe;
    p_tab->adr_s_dic=0;
    p_tab->nbr_s_rub=0;
    p_tab->i_tab    =i_tab++;
}

copier_sup(niv)
short niv;
{
    short niv_a_copier;
    short i_a_copier;
    short n_a_copier;

```


#####	#####	#	#	#####	#####	#####	#####
#	#	#	##	##	#	#	#
#	#	#####	#	##	#	#	#####
#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#
#####	#####	#	#	#####	#	#	#####

Tue May 17 11:23:10 GMT 1988

井井井井井	井井井井井井井	井	井	井井井井井	井井井井井井井	井井井井井井井	井井井井井井井井井
井	井	井	井井	井井	井	井	井
井	井	井井井井井井	井	井井	井	井	井井井井井井
井	井	井	井	井	井	井	井
井	井	井	井	井	井	井	井
井井井井井井	井井井井井井井	井	井	井井井井井	井	井	井井井井井井井井

Tue May 17 11:23:30 GMT 1988

/*

Directory : /usr2/demotte

Intitule : corps1.c

Compile : cc corps1.c ecran1.c -les -o et3.exe

Regroupe toutes les procedures autres que les procedures dites d'ecran
du programme charge de la mise a jour du fichier de la premiere version
encodée du repertoire de KENT par insertion d'elements tires du fichier
de la version encodée du repertoire de BOENNINGHAUSEN.

*/

#include "/usr/archimed/lib/iodef.c"

#define NBR_FLE 14

#include "/usr/archimed/lib/iodec.c"

#include "/usr2/demotte/def.c"

FILE *fopen(), *fgroupe;

int f_groupe;

char * nom_bce;

char * nom_d1_boe;

char * nom_d2_boe;

char * nom_d3_boe;

char * nom_dn_boe;

char * nom_groupe;

char * nom_medco;

char * nom_ken;

char * nom_d1_ken;

char * nom_d2_ken;

char * nom_d3_ken;

char * nom_dn_ken;

char * nom_rub;

char * nom_kenbis;

char * nom_inter;

char boe[_WND_LGT];

struct boe123 d1_boe[DIM_D1_BOE];

struct boe123 d2_boe[DIM_D2_BOE];

struct boe123 d3_boe[DIM_D3_BOE];

struct boen dn_boe[DIM_DN_BOE];

char medco[LGR_MEDCO];

char ken[_WND_LGT];

struct ken123 d1_ken[DIM_D1_KEN];

struct ken123 d2_ken[DIM_D2_KEN];

struct ken123 d3_ken[DIM_D3_KEN];

struct kenn dn_ken[DIM_DN_KEN];

char rub[_WND_LGT];

char kenbis[_WND_LGT];

struct des inter[DIM_INTER];

struct nemo nemob[NIV_MAX+1];

struct nemo nemok[NIV_MAX+1];

struct ordre * deb_ordre;

struct ordre * p_ordre;

struct ORDRE * deb_ORDRE;

struct ORDRE * p_ORDRE;

struct liste * deb_liste;

struct liste * p_liste;

struct rep * deb_rep1;

struct rep * p_rep1;


```
struct rep * deb_rep1;
struct rep * p_rep1;
struct rep * deb_rep2;
struct rep * p_rep2;
struct rep * deb_rep3;
struct rep * p_rep3;
```

```
struct des * deb_des;
struct des * p_des;
```

```
struct med * deb_med;
struct med * p_med;
struct med * p_medprec;
```

```
struct boe123 * p_boe123;
struct boen * p_boen;
struct ken123 * p_ken123;
struct kenn * p_kenn;
```

```
char cle[L_CLE];
```

```
short mode;
short nivb;
short nivk;
short max_niv;
char intit_fin[512];
char type;
long adr_deb;
long i_boe;
long i_groupe;
long i_ken;
long i_medco;
long adr[NIV_MAX+1];
short modif[NIV_MAX+1];
long adr_groupe;
short niv_deb;
char ret;
long l_boe;
long l_rub;
long l_mecco;
long l_ker;
short pos;
short der_riv;
char symptome[INTITS_PAGE*L_ECRAN];
/*
```

```

*/
main()
{
char car;

B_clear(_SCREEN, _ALL, 0, 0, 0);
B_put_screen();
system("stty raw -echo");
prologue();
for(ecran1(1), car=acquerir( -1, 1); car!='a'; ecran1(1), car=acquerir( -1, 1))
    switch(car)
    {
        case 'i':if((definir_deb())&&
                    (definir_fin())&&
                    (definir_insert()))
                {
                    traiter_groupe();
                    representer();
                };
                break;
        case 'f':if((definir_liste1())&&
                    (definir_liste2()))
                {
                    traiter_liste();
                    representer();
                };
                break;
        default :break;
    };

for(ecran1(2), car=acquerir( -1, 1); (car!='r')&&(car!='i'); car=acquerir( -1, 1));
B_clear(_SCREEN, _ALL, 0, 0, 0);
B_put_screen();
system("stty 9600 -raw echo cs8 -parenb -istrip");
switch(car)
{
case 'r':/*test();*/
        epilogue(1);
        break;
case 'i':/*test();*/
        epilogue(2);
        inserer_groupes();
        printf("\n max niv= %d",max_niv);
        epilogue(3);
        break;
};
}

```

/*

```

*/
prologue()

```



```

*/
prologue()

```

```

{
    char car;
    char * s;
    short i;
    short index;
    struct des * p_desprec;

```

```

    nom_boe      = "/usr2/demotte/BOE.DAT";
    nom_d1_boe   = "/usr2/demotte/D1_BOE.DAT";
    nom_d2_boe   = "/usr2/demotte/D2_BOE.DAT";
    nom_d3_boe   = "/usr2/demotte/D3_BOE.DAT";
    nom_dn_boe   = "/usr2/demotte/DN_BOE.DAT";
    nom_groupe   = "/usr2/demotte/GROUPE.DAT";
    nom_medco    = "/usr2/demotte/BKA.DAT";
    nom_ken      = "/usr/archimed/kent/rep1.dat";
    nom_d1_ken   = "/usr/archimed/kent/DIC1.DAT";
    nom_d2_ken   = "/usr/archimed/kent/DIC2.DAT";
    nom_d3_ken   = "/usr/archimed/kent/DIC3.DAT";
    nom_dn_ken   = "/usr/archimed/kent/DICN.DAT";
    nom_rub      = "/usr/archimed/kent/RUB.DAT";
    nom_kenbis   = "/usr2/demotte/KENbis.DAT";
    nom_inter    = "/usr2/demotte/INTER.DAT";

```

```

_def_file(_FILE_NAME_IS      nom_boe
         _FILE_IDENTIFIER_IS BOE
         _BUFFER_IS          boe
         _WINDOW_LENGTH_IS   _WND_LGT
         _RECORD_LENGTH_IS   1
         _ORGANISATION_IS    _SEQ
         _DENSITY_IS         _SGL
         _WHEN_EOF_USE       0
         _WHEN_ERROR_USE     0
         _FOR_TRACING_USE    0);

```

```

_def_file(_FILE_NAME_IS      nom_d1_boe
         _FILE_IDENTIFIER_IS D1_BOE
         _BUFFER_IS          d1_boe
         _WINDOW_LENGTH_IS   LGR_D1_BOE
         _RECORD_LENGTH_IS   ART_D1_BOE
         _ORGANISATION_IS    _SEQ
         _DENSITY_IS         _SGL
         _WHEN_EOF_USE       0
         _WHEN_ERROR_USE     0
         _FOR_TRACING_USE    0);

```

```

_def_file(_FILE_NAME_IS      nom_d2_boe
         _FILE_IDENTIFIER_IS D2_BOE
         _BUFFER_IS          d2_boe
         _WINDOW_LENGTH_IS   LGR_D2_BOE
         _RECORD_LENGTH_IS   ART_D2_BOE
         _ORGANISATION_IS    _SEQ
         _DENSITY_IS         _SGL
         _WHEN_EOF_USE       0
         _WHEN_ERROR_USE     0
         _FOR_TRACING_USE    0);

```

```

_def_file(_FILE_NAME_IS      nom_d3_boe

```

_FILE_IDENTIFIER_IS	D3_BOE
_BUFFER_IS	d3_boe
_WINDOW_LENGTH_IS	LGR_D3_BOE
_RECORD_LENGTH_IS	ART_D3_BOE
_ORGANISATION_IS	_SEQ
_DENSITY_IS	_SGL
_WHEN_EOF_USE	0
_WHEN_ERROR_USE	0
_FOR_TRACING_USE	0);

_def_file(_FILE_NAME_IS	nom_dn_boe
_FILE_IDENTIFIER_IS	DN_BOE
_BUFFER_IS	dn_boe
_WINDOW_LENGTH_IS	LGR_DN_BOE
_RECORD_LENGTH_IS	ART_DN_BOE
_ORGANISATION_IS	_SEQ
_DENSITY_IS	_SGL
_WHEN_EOF_USE	0
_WHEN_ERROR_USE	0
_FOR_TRACING_USE	0);

_def_file(_FILE_NAME_IS	nom_medco
_FILE_IDENTIFIER_IS	MEDCO
_BUFFER_IS	medco
_WINDOW_LENGTH_IS	LGR_MEDCO
_RECORD_LENGTH_IS	1
_ORGANISATION_IS	_SEQ
_DENSITY_IS	_SGL
_WHEN_EOF_USE	0
_WHEN_ERROR_USE	0
_FOR_TRACING_USE	0);

_def_file(_FILE_NAME_IS	nom_ken
_FILE_IDENTIFIER_IS	KEN
_BUFFER_IS	ken
_WINDOW_LENGTH_IS	_WND_LGT
_RECORD_LENGTH_IS	1
_ORGANISATION_IS	_SEQ
_DENSITY_IS	_SGL
_WHEN_EOF_USE	0
_WHEN_ERROR_USE	0
_FOR_TRACING_USE	0);

_def_file(_FILE_NAME_IS	nom_d1_ken
_FILE_IDENTIFIER_IS	D1_KEN
_BUFFER_IS	d1_ken
_WINDOW_LENGTH_IS	LGR_D1_KEN
_RECORD_LENGTH_IS	ART_D1_KEN
_ORGANISATION_IS	_SEQ
_DENSITY_IS	_SGL
_WHEN_EOF_USE	0
_WHEN_ERROR_USE	0
_FOR_TRACING_USE	0);

_def_file(_FILE_NAME_IS	nom_d2_ken
_FILE_IDENTIFIER_IS	D2_KEN
_BUFFER_IS	d2_ken
_WINDOW_LENGTH_IS	LGR_D2_KEN
_RECORD_LENGTH_IS	ART_D2_KEN

_ORGANISATION_IS	_SEQ
_DENSITY_IS	_SGL
_WHEN_EOF_USE	0

_ORGANISATION_IS	_SEQ
_DENSITY_IS	_SGL
_WHEN_EOF_USE	0
_WHEN_ERROR_USE	0
_FOR_TRACING_USE	0);
_def_file(_FILE_NAME_IS	nom_d3_ken
_FILE_IDENTIFIER_IS	D3_KEN
_BUFFER_IS	d3_ken
_WINDOW_LENGTH_IS	LGR_D3_KEN
_RECORD_LENGTH_IS	ART_D3_KEN
_ORGANISATION_IS	_SEQ
_DENSITY_IS	_SGL
_WHEN_EOF_USE	0
_WHEN_ERROR_USE	0
_FOR_TRACING_USE	0);
_def_file(_FILE_NAME_IS	nom_dn_ken
_FILE_IDENTIFIER_IS	DN_KEN
_BUFFER_IS	dn_ken
_WINDOW_LENGTH_IS	LGR_DN_KEN
_RECORD_LENGTH_IS	ART_DN_KEN
_ORGANISATION_IS	_SEQ
_DENSITY_IS	_SGL
_WHEN_EOF_USE	0
_WHEN_ERROR_USE	0
_FOR_TRACING_USE	0);
_def_file(_FILE_NAME_IS	nom_rub
_FILE_IDENTIFIER_IS	RUB
_BUFFER_IS	rub
_WINDOW_LENGTH_IS	_WND_LGT
_RECORD_LENGTH_IS	1
_ORGANISATION_IS	_SEQ
_DENSITY_IS	_SGL
_WHEN_EOF_USE	0
_WHEN_ERROR_USE	0
_FOR_TRACING_USE	0);
_def_file(_FILE_NAME_IS	nom_kenbis
_FILE_IDENTIFIER_IS	KENbis
_BUFFER_IS	kenbis
_WINDOW_LENGTH_IS	_WND_LGT
_RECORD_LENGTH_IS	1
_ORGANISATION_IS	_SEQ
_DENSITY_IS	_SGL
_WHEN_EOF_USE	0
_WHEN_ERROR_USE	0
_FOR_TRACING_USE	0);
_def_file(_FILE_NAME_IS	nom_inter
_FILE_IDENTIFIER_IS	INTER
_BUFFER_IS	inter
_WINDOW_LENGTH_IS	LGR_INTER
_RECORD_LENGTH_IS	ART_INTER
_ORGANISATION_IS	_SEQ
_DENSITY_IS	_SGL
_WHEN_EOF_USE	0
_WHEN_ERROR_USE	0
_FOR_TRACING_USE	0);

```

_opn_fle(BOE _IN_MODE _INP);
_opn_fle(D1_BOE _IN_MODE _INP);
_opn_fle(D2_BOE _IN_MODE _INP);
_opn_fle(D3_BOE _IN_MODE _INP);
_opn_fle(DN_BOE _IN_MODE _INP);
_opn_fle(MEDCO _IN_MODE _INP);
_opn_fle(KEN _IN_MODE _INP);
_opn_fle(D1_KEN _IN_MODE _INP);
_opn_fle(D2_KEN _IN_MODE _INP);
_opn_fle(D3_KEN _IN_MODE _INP);
_opn_fle(DN_KEN _IN_MODE _INP);
_opn_fle(RUB _IN_MODE _INP);
_opn_fle(KENbis _IN_MODE _OUT);

fgroupe=fopen(nom_groupe,"a");

_opn_fle(INTER _IN_MODE _INP);
deb_des=(struct des *)malloc(sizeof(struct des));
deb_des->suivant=0;
for(i=1; i<=NIV_MAX; i++) deb_des->lock[i]= -1;
p_des=deb_des;
_red_fle(INTER, _BYT, LGR_INTER, (index=0)*ART_INTER);
while(inter[index%DIM].nivk!= -1)
{
    p_desprec=p_des;
    p_desprec->suivant=(struct des *)malloc(sizeof(struct des));
    p_des=p_desprec->suivant;
    for(i=1; i<=NIV_MAX; i++)
        p_des->lock[i]=inter[index%DIM].lock[i];
    p_des->typk =inter[index%DIM].typk;
    p_des->nivk =inter[index%DIM].nivk;
    p_des->adrg =inter[index%DIM].adrg;
    p_des->suivant=0;
    if(++index%DIM==0) _red_fle(INTER, _BYT, LGR_INTER, index*ART_INTER);
};
i_groupe=inter[index%DIM].lock[0];
max_niv =inter[index%DIM].lock[1];
_cls_fle(INTER);

_siz(BOE,&l_boe);
_siz(RUB,&l_rub);
_siz(KEN,&l_ken);
_siz(MEDCO,&l_medco);

/* boe>kent */
deb_rep1=(struct rep *)malloc(sizeof(struct rep));
deb_rep1->suivant=0;
deb_rep1->car =SPACE;
p_rep1=deb_rep1;
for(i_medco=0; i_medco<l_medco-2;)
{
    if((car=get_chr(1))!=p_rep1->car)
    {
        p_rep1->suivant=(struct rep *)malloc(sizeof(struct rep));
        p_rep1=p_rep1->suivant;
        p_rep1->car =car;
        p_rep1->loc =i_medco-1;
        p_rep1->suivant=0;
    };
};

while((car=get_chr(1))!=_CRT);
};

```



```

    while((car=get_chr(1))!=_CRT);
};

/* abre>kent */
_red_fle(MEDCO, _BYT, LGR_MEDCO, 0);
deb_rep2=(struct rep *)malloc(sizeof(struct rep));
deb_rep2->suivant=0;
deb_rep2->car=SPACE;
p_rep2=deb_rep2;
for(i=0,i_medco=0; (car=get_chr(1))!='*'; );
for( ; (car=get_chr(1))!='*'; );
while(1)
{
    if((car=get_chr(1))!=p_rep2->car)
    {
        p_rep2->suivant=(struct rep *)malloc(sizeof(struct rep));
        p_rep2=p_rep2->suivant;
        p_rep2->car=car;
        p_rep2->loc=i;
        p_rep2->suivant=0;
    };
    while((car=get_chr(1))!=_CRT);
    if((i=i_medco)>=1_medco-2) break;
    while((car=get_chr(1))!='*');
    while((car=get_chr(1))!='*');
};

/* kent>abre */
_red_fle(MEDCO, _BYT, LGR_MEDCO, 0);
deb_rep3=(struct rep *)malloc(sizeof(struct rep));
deb_rep3->suivant=0;
deb_rep3->car=SPACE;
p_rep3=deb_rep3;
for(i_medco=0; (car=get_chr(1))!='*');
while(1)
{
    if((car=get_chr(1))!=p_rep3->car)
    {
        p_rep3->suivant=(struct rep *)malloc(sizeof(struct rep));
        p_rep3=p_rep3->suivant;
        p_rep3->car=car;
        p_rep3->loc=i_medco-1;
        p_rep3->suivant=0;
    };
    while((car=get_chr(1))!=_CRT);
    if(i_medco>=1_medco-2) break;
    while((car=get_chr(1))!='*');
};

nemob[1].adr_dic=0;
nemob[1].lgr_dic=LGR_DIC1B;
nemob[1].nbr_rub=NBR_DIC1B;
nemob[1].adr=-1;

nemok[1].adr_dic=0;
nemok[1].lgr_dic=LGR_DIC1K;
nemok[1].nbr_rub=NBR_DIC1K;
nemok[1].adr=-1;

```

```

for(i=2; i<=NIV_MAX; i++)
{
    nemob[i].adr_dic= -1;
    nemob[i].lgr_dic= -1;
    nemob[i].nbr_rub= -1;
    nemob[i].adr      = -1;

    nemok[i].adr_dic= -1;
    nemok[i].lgr_dic= -1;
    nemok[i].nbr_rub= -1;
    nemok[i].adr      = -1;
};

deb_ordre=(struct ordre *)malloc(sizeof(struct ordre));
deb_ordre->suivant=0;

deb_ORDRE=(struct ORDRE *)malloc(sizeof(struct ORDRE));
deb_ORDRE->suivant=0;

deb_liste=(struct liste *)malloc(sizeof(struct liste));
deb_liste->suivant=0;
s="etc";
for(i=0; s[i]!='\0'; i++)
    deb_liste->corr[i]=deb_liste->abre[i]=s[i];
deb_liste->corr[i]=deb_liste->abre[i]='\0';

deb_med=(struct med *)malloc(sizeof(struct med));
deb_med->suivant=0;
}

```

```

get_chr(cas)
short cas;
{
    char car;

    switch(cas)
    {
        case 1:
            i_medco++;
            return(car=_get_nxt_chr(MEDCO));

        case 2:
            i_boe++;
            return(car=_get_nxt_chr(BOE));

        case 3:
            i_ken++;
            return(car=_get_nxt_chr(KEN));
    };
}

```

```

test()
{
    char car;
    short i;
    short j;

    j=0;

```

```

p_des=deb_des;
while(p_des->suivant!=0)
    r

```



```

p_des=deb_des;
while(p_des->suiwant!=0)
{
    B_locate(++j,1);
    B_put_screen();
    p_des=p_des->suiwant;
    for(i=1; i<=NIV_MAX; i++)
    {
        if(p_des->lock[i]== -1)
        {
            i--;
            break;
        };
        printf(">");
        if(p_des->lock[i]<l_rub-2)
        {
            _red_fle(RUB, _BYT, _WND_LGT, p_des->lock[i]);
            while((car=_get_nxt_chr(RUB))!=';')
                printf("%c",car);
        };
    };
    B_locate(++j,3);
    B_put_screen();
    printf("typ=%c niv=%d adr=%d adrg=%d",
    p_des->typk,p_des->nivk,p_des->lock[i],p_des->adrg);
    scanf("%c",&ret);
};

```

/*

```

*/
definir_deb()
{
    short i;

    nivb = 1;
    if(i==mode_i1())
    {
        niv_deb=nivb;
        adr_deb=(nivb<=3 ? p_boe123->adr_boe : p_boen->adr_boe);
    };
    return(i);
}

```

```

definir_fin()
{
    short i;
    short j;
    char car;

    if(i==mode_i2())
        if(i==((niv_deb<=3 ? p_boe123->adr_boe : p_boen->adr_boe)>=adr_deb))
        {
            _red_fle(BOE, _BYT, _WND_LGT, (niv_deb<=3 ? p_boe123->adr_boe : p_boen->adr_boe));
            for(j=0,car=_get_nxt_chr(BOE); car!=': '; j++,car=_get_nxt_chr(BOE))
                intit_fin[j]=car;
            intit_fin[j]=': ';
        };
    return(i);
}

```

```

definir_insert()
{
    nivk = 1;
    return(mode_i3());
}

```

```

definir_liste1()
{
    short i;
    char car;

    nivb=1;
    if(i==mode_i1())
    {
        adr_deb=(nivb<=3 ? p_boe123->adr_boe : p_boen->adr_boe);
        _red_fle(BOE, _BYT, _WND_LGT, (i_boe=adr_deb));
        for(i=0; i_boe<l_boe-2; )
        {
            car=get_chr(2);
            if((car=='&')||(car=='+' )||(car=='*'))
            {
                i=(car=='*' ? 1 : 0);
                break;
            }
        }
    }
}

```



```

    };
    };
    return(i);
}

```

```

definir_liste2()
{
    short i;

    nivk=1;
    if(i=mode_i3())
        if(i=(type=='E' ? 1 : 0)) type='*';
    return(i);
}

```

```

mode_n()
{
    char car;

    ecran2( 1, (mode==3 ? nivk : nivb), 0);
    while(1)
    switch(car=acquerir( -1, mode%2))
    {
    case _ESC:if((car=getchar())=='[')
        switch(car=getchar())
        {
        case 'C' :
            if(((mode==3 ? nivk : nivb)<NIV_MAX)&&
                ((mode==3 ? nemok[nivk].adr : nemob[nivb].adr)!= -1))
            {
                visionner_intit('O',(mode==3 ? nivk : nivb),0);
                visionner_niv(mode==3 ? ++nivk : ++nivb);
            };
            break;

        case 'D' :
            if((mode==3 ? nivk : nivb)>1)
            {
                visionner_niv(mode==3 ? --nivk : --nivb);
                visionner_blanc(mode==3 ? nivk : nivb);
            };
            break;

        case 'B' :
            return(mode==3 ? mode_i3() : mode_i1());

        default:
            break;
        };
        break;

    default :break;
    };
}

```

```

mode_i1()
{
    char car;
    char intit[512];
    short i;
    short j;

    mode=1;
    ecran2(2, nivb, 0);
    for(i= -1, j=0;;)
        switch(car=acquerir(i,1))
        {
            case _DEL :if(i>=0) i--;
                        break;

            case '/' :
                        if(j)
                        {
                            visionner_blanc(nivb);
                            nivb--;
                        };
                        return(j);

            case _ESC :if((car=getchar())=='[')
                        switch(car=getchar())
                        {
                            case 'A':return(mode_n());
                            default :break;
                        };
                        break;

            case _ret :intit[++i]='\0';
                        if(j=chercher_i12(intit,1)) visionner_niv(++nivb);
                        i= -1;
                        break;

            default :if(_UPR_CAS_CHR(car)) car+=32;
                        intit[++i]=car;
                        break;

        };
}

```

```

mode_i2()
{
    char car;
    char intit[512];
    short i;
    short j;

    mode=2;
    ecran2(3, nivb, 0);
    for(i= -1, j=0;;)
        switch(car=acquerir(i,0))
        {
            case _DEL :if(i>=0) i--;
                        break;

            case '/' :return(j);

            case _ret :intit[++i]='\0';

```



```

        case '/' :return(j);

        case _ret :intit[++i]='\0';
                    j=chercher_i12(intit,0);
                    i= -1;
                    break;

        default :if(_UPR_CAS_CHR(car)) car+=32;
                    intit[++i]=car;
                    break;
    };
}

```

```

mode_i3()
{
    char car;
    char intit[512];
    short i;
    short j;

    mode=3;
    ecran2(4, nivk, 0);
    for(i= -1,j=0;;)
        switch(car=acquerir(i,1))
        {
            case _DEL :if(i>=0) i--;
                        break;

            case '/' :if(j) nivk--;
                        return(j);

            case _ESC :if((car=getchar())=='[')
                            switch(car=getchar())
                            {
                                case 'A':return(mode_n());
                                default :break;
                            };
                        break;

            case _ret :intit[++i]='\0';
                        if(j=chercher_i3(intit)) visionner_niv(++nivk);
                        i= -1;
                        break;

            default :if(_UPR_CAS_CHR(car)) car+=32;
                        intit[++i]=car;
                        break;
        };
}

```

```

chercher_i12(intit,i)
char intit[];
short i;
{

```

```

short n_repere;
short u;
short v;

switch(n_repere=intitule_repere(intit))
{
case 0: visionner_intit('Z',nivb,0);
        return(0);
case 1: mise_a_jourb(0, 0);
        visionner_intit('D',nivb,0);
        return(1);
default: u=1;
         v=1;
         selectionner(n_repere, &u, &v);
         ecran2((i ? 2 : 3), nivb, 0);
         mise_a_jourb(u, v);
         visionner_intit('D',nivb,0);
         return(1);
};
}

```

```

chercher_i3(intit)
) char intit[];
) {
) short n_repere;
) short u;
) short v;

) switch(intit[0])
) {
) case '.' : switch(nemok[nivk].lgr_dic)
) {
) case 0 : type='N';
) visionner_intit('N',nivk-1,0);
) return(1);
) default: type='F';
) visionner_intit('L',nivk,0);
) return(1);
) };
) default : switch(n_repere=intitule_repere(intit))
) {
) case 0 : visionner_intit('Z',nivk,0);
) return(0);
) case 1 : mise_a_jourk(0, 0);
) type='E';
) visionner_intit('D',nivk,0);
) return(1);
) default: u=1;
) v=1;
) selectionner(n_repere, &u, &v);
) ecran2(4, nivk, 0);
) mise_a_jourk(u, v);
) type='E';
) visionner_intit('D',nivk,0);
) return(1);
) };
);
}

```



```

intitule_repere(intit)
{
    char intit[];
    {
        short i;
        short j;
        short n_repere;
        char * ptr;
        char car;

        mise_a_neuf();
        if((mode==3 ? nivk : nivb)<=3)
        {
            charger_cle((mode==3 ? nivk : nivb),
                (mode==3 ?
                    nemok[nivk].adr_dic+nemok[nivk].nbr_rub*_RCD_DEF(D1_KEN+nivk-1) :
                    nemob[nivb].adr_dic+nemob[nivb].nbr_rub*_RCD_DEF(D1_BOE+nivb-1)),
                (mode==3 ?
                    nemok[nivk].lgr_dic-nemok[nivk].nbr_rub*_RCD_DEF(D1_KEN+nivk-1) :
                    nemob[nivb].lgr_dic-nemob[nivb].nbr_rub*_RCD_DEF(D1_BOE+nivb-1)));
            n_repere=0;
            for(i=0; i<(mode==3 ? nemok[nivk].nbr_rub : nemob[nivb].nbr_rub); i++)
            {
                if(i%DIM==0)
                    charger_dic((mode==3 ? nivk : nivb),
                        i/DIM,
                        (mode==3 ? nemok[nivk].adr_dic : nemob[nivb].adr_dic),
                        (mode==3 ? nemok[nivk].nbr_rub : nemob[nivb].nbr_rub));

                if(mode==3)
                    p_ken123=(struct ken123 *)(_BUF(D1_KEN+nivk-1)+(i%DIM)*_RCD_DEF(D1_KEN+nivk-1));
                else
                    p_boe123=(struct boe123 *)(_BUF(D1_BOE+nivb-1)+(i%DIM)*_RCD_DEF(D1_BOE+nivb-1));

                for(j=0, ptr=cle+(mode==3 ? p_ken123->ptr : p_boe123->ptr); j++,ptr++)
                {
                    while(intit[j]==SPACE) j++;
                    while(*ptr==SPACE) ptr++;
                    if((intit[j]=='\0')||(*ptr=='\0'))
                    {
                        n_repere++;
                        ajouter(mode==3 ? p_ken123->n_ordre : p_boe123->n_ordre);
                        break;
                    };
                    if(intit[j]> *ptr) break;
                    if(intit[j]< *ptr) return(n_repere);
                };
            };
            return(n_repere);
        }
    }
else
{
    n_repere=0;
    for(i=0; i<(mode==3 ? nemok[nivk].nbr_rub : nemob[nivb].nbr_rub); i++)
    {
        if(i%DIM==0)
            charger_dic((mode==3 ? nivk : nivb),
                i/DIM,

```

```

(mode==3 ? nemok[nivk].adr_dic : nemob[nivb].adr_dic),
(mode==3 ? nemok[nivk].nbr_rub : nemob[nivb].nbr_rub));

if(mode==3)
p_kenn=(struct kenn *)(_BUF(DN_KEN)+(i%DIM)*_RCD_DEF(DN_KEN));
else
p_boen=(struct boen *)(_BUF(DN_BOE)+(i%DIM)*_RCD_DEF(DN_BOE));

_red_fle((mode==3 ? RUB : BOE), _BYT, _WND_LGT, (mode==3 ? p_kenn->adr_rub+2 : p_boen->a
for(j=0,car=_get_nxt_chr(mode==3 ? RUB : BOE));
j++,car=_get_nxt_chr(mode==3 ? RUB : BOE))
{
while(intit[j]==SPACE) j++;
while((car==SPACE)!!(car=='[')!!(car==_CRT))
car=_get_nxt_chr(mode==3 ? RUB : BOE);
if(_UPR_CAS_CHR(car)) car +=32;
if((intit[j]=='\0')!!(car==' '))
{
n_repere++;
ajouter(i);
break;
};
if(intit[j]>car) break;
if(intit[j]<car) break;
};
};
return(n_repere);
};
}

```

```

charger_dic(niv,i,adr_dic,nbr_rub)
short niv;
short i;
long adr_dic;
short nbr_rub;
{
switch(niv)
{
case 1 :
case 2 :
case 3 : _red_fle((mode/3 ? D1_KEN : D1_BOE)+niv-1, _BYT, (nbr_rub-i%DIM<DIM ?
(nbr_rub-i%DIM)*_RCD_DEF((mode/3 ? D1_KEN : D1_BOE)+niv-1) :
DIM*_RCD_DEF((mode/3 ? D1_KEN : D1_BOE)+niv-1)),
adr_dic+i%DIM*_RCD_DEF((mode/3 ? D1_KEN : D1_BOE)+niv-1));
break;
default: _red_fle((mode/3 ? DN_KEN : DN_BOE), _BYT, (nbr_rub-i%DIM<DIM ?
(nbr_rub-i%DIM)*_RCD_DEF(mode/3 ? DN_KEN : DN_BOE) :
DIM*_RCD_DEF(mode/3 ? DN_KEN : DN_BOE)),
adr_dic+i%DIM*_RCD_DEF(mode/3 ? DN_KEN : DN_BOE));
break;
};
}

```

```

charger_cle(niv,adr,lgr)
short niv;
long adr;
long lgr;

```

```

{
char * adr_buf;

```

```

adr_buf= BUF((mode/3 ? D1_KEN : D1_BOE)+niv-1);

```



```

{
char * adr_buf;

adr_buf=_BUF((mode/3 ? D1_KEN : D1_BOE)+niv-1);
_BUF((mode/3 ? D1_KEN : D1_BOE)+niv-1)=cle;
_red_fle((mode/3 ? D1_KEN : D1_BOE)+niv-1, _BYT, lgr, adr);
_BUF((mode/3 ? D1_KEN : D1_BOE)+niv-1)=adr_buf;
}

```

```

ajouter(n_ordre)
short n_ordre;
{
p_ordre=deb_ordre;
while(p_ordre->suivant!=0) p_ordre=p_ordre->suivant;
p_ordre->suivant=(struct ordre *)malloc(sizeof(struct ordre));
p_ordre=p_ordre->suivant;
p_ordre->n_ordre=n_ordre;
p_ordre->suivant=0;
}

```

```

mise_a_jourk(u, v)
short u;
short v;
{
short i;

if((u==0)&&(v==0)) p_ordre=deb_ordre->suivant;
else
{
for(i=1, p_ORDRE=deb_ORDRE->suivant; i<u; i++, p_ORDRE=p_ORDRE->suivant);
for(i=1, p_ordre=p_ORDRE->p_ordre; i<v; i++, p_ordre=p_ordre->suivant);
};
charger_dic(nivk,
            p_ordre->n_ordre/DIM,
            nemok[nivk].adr_dic,
            nemok[nivk].nbr_rub);
if(nivk<=2)
{
p_ken123=(struct ken123 *)
(_BUF(D1_KEN+nivk-1)+(p_ordre->n_ordre%DIM)*_RCD_DEF(D1_KEN+nivk-1));
nemok[nivk].adr      =p_ken123->adr_rub;
nemok[nivk+1].adr_dic=p_ken123->adr_s_dic;
nemok[nivk+1].lgr_dic=p_ken123->lgr_s_dic;
nemok[nivk+1].nbr_rub=p_ken123->nbr_s_rub;
}
else
{
if(nivk==3)
{
p_ken123=(struct ken123 *)
(_BUF(D3_KEN)+(p_ordre->n_ordre%DIM)*_RCD_DEF(D3_KEN));
nemok[3].adr      =p_ken123->adr_rub;
nemok[4].adr_dic=(p_ken123->adr_s_dic+p_ken123->lgr_s_dic-p_ken123->nbr_s_rub)*_RCD_DEF(DN_KEN);
nemok[4].lgr_dic=p_ken123->lgr_s_dic;
nemok[4].nbr_rub=p_ken123->nbr_s_rub;
}
}
}

```

```

else
{
if (3<nivk<NIV_MAX)
{
p_kenn=(struct kenn *)
(_BUF(DN_KEN)+(p_ordre->n_ordre%DIM)*_RCD_DEF(DN_KEN));
nemok[nivk].adr = p_kenn->adr_rub;
nemok[nivk+1].adr_dic=nemok[4].adr_dic+
(p_kenn->adr_s_dic-(nemok[4].lgr_dic-nemok[4].nbr_rub))*_RCD_DEF(DN_KEN);
nemok[nivk+1].lgr_dic=p_kenn->lgr_s_dic;
nemok[nivk+1].nbr_rub=p_kenn->lgr_s_dic;
}
else
{
p_kenn=(struct kenn *)
(_BUF(DN_KEN)+(p_ordre->n_ordre%DIM)*_RCD_DEF(DN_KEN));
nemok[nivk].adr=p_kenn->adr_rub;
};
};
};

for(i=nivk+2; i<=NIV_MAX; i++)
{
nemok[i].adr = -1;
nemok[i].adr_dic= -1;
nemok[i].lgr_dic= -1;
nemok[i].nbr_rub= -1;
};
}

```

```

mise_a_jourb(u, v)
short u;
short v;
{
short i;

if((u==0)&&(v==0)) p_ordre=deb_ordre->suivant;
else
{
for(i=1, p_ORDRE=deb_ORDRE->suivant; i<u; i++, p_ORDRE=p_ORDRE->suivant);
for(i=1, p_ordre=p_ORDRE->p_ordre; i<v; i++, p_ordre=p_ordre->suivant);
};
charger_dic(nivb,
p_ordre->n_ordre/DIM,
nemob[nivb].adr_dic,
nemob[nivb].nbr_rub);
if(nivb<=2)
{
p_boe123=(struct boe123 *)
(_BUF(D1_BOE+nivb-1)+(p_ordre->n_ordre%DIM)*_RCD_DEF(D1_BOE+nivb-1));
nemob[nivb].adr = p_boe123->adr_boe;
nemob[nivb+1].adr_dic=p_boe123->adr_s_dic;
nemob[nivb+1].lgr_dic=p_boe123->lgr_s_dic;
nemob[nivb+1].nbr_rub=p_boe123->nbr_s_rub;
}
else
{
if(nivb==3)

```

```

{
p_boe123=(struct boe123 *)
(_BUF(D1_BOE+nivb-1)+(p_ordre->n_ordre%DIM)*_RCD_DEF(D1_BOE+nivb-1));

```



```

    {
        p_boe123=(struct boe123 *)
        (_BUF(D3_BOE)+(p_ordre->n_ordre%DIM)*_RCD_DEF(D3_BOE));
        nemob[nivb].adr      =p_boe123->adr_boe;
        nemob[nivb+1].adr_dic=p_boe123->adr_s_dic+
        (p_boe123->lgr_s_dic-p_boe123->nbr_s_rub)*_RCD_DEF(DN_BOE);
        nemob[nivb+1].lgr_dic=p_boe123->lgr_s_dic;
        nemob[nivb+1].nbr_rub=p_boe123->nbr_s_rub;
    }

else
{
    if(3<nivb<NIV_MAX)
    {
        p_boen=(struct boen *)
        (_BUF(DN_BOE)+(p_ordre->n_ordre%DIM)*_RCD_DEF(DN_BOE));
        nemob[nivb].adr      =p_boen->adr_boe;
        nemob[nivb+1].adr_dic=nemob[4].adr_dic+
        (p_boen->adr_s_dic-(nemob[4].lgr_dic-nemob[4].nbr_rub))
        *_RCD_DEF(DN_BOE);
        nemob[nivb+1].nbr_rub=p_boen->nbr_s_rub;
    }

else
    p_boen=(struct boen *)
    (_BUF(DN_BOE)+(p_ordre->n_ordre%DIM)*_RCD_DEF(DN_BOE));
    nemob[nivb].adr=p_boen->adr_boe;
};
};

for(i=nivb+2; i<=NIV_MAX; i++)
{
    nemob[i].adr      = -1;
    nemob[i].adr_dic= -1;
    nemob[i].lgr_dic= -1;
    nemob[i].nbr_rub= -1;
};
}

```

mise_a_neuf()

```

{
    p_ORDRE=deb_ORDRE;
    while(p_ORDRE->suivant!=0)
    {
        p_ORDRE=p_ORDRE->suivant;
        free(p_ORDRE);
    };
    deb_ORDRE->suivant=0;
    p_ORDRE=deb_ORDRE;

    p_ordre=deb_ordre;
    while(p_ordre->suivant!=0)
    {
        p_ordre=p_ordre->suivant;
        free(p_ordre);
    };
    deb_ordre->suivant=0;
    p_ordre=deb_ordre;
}

```

```

selectionner(n_repere, p_u, p_v)
    short n_repere;
    short * p_u;
    short * p_v;
    {
        short i;
        short max_u;
        short min_v;
        char car;

        p_ordre=deb_ordre->suiwant;
        p_ORDRE=deb_ORDRE;
        for(i=0; i>=0 ;)
        {
            p_ORDRE->suiwant=(struct ORDRE *)malloc(sizeof(struct ORDRE));
            p_ORDRE=p_ORDRE->suiwant;
            p_ORDRE->p_ordre =p_ordre;
            p_ORDRE->suiwant =0;
            if(p_ordre->suiwant==0) break;
            p_ordre=p_ordre->suiwant;
            i++;
            while(i%INTITS_PAGE!=0)
            {
                if(p_ordre->suiwant==0)
                {
                    i= -1;
                    break;
                }
                p_ordre=p_ordre->suiwant;
                i++;
            }
        }
    }

max_u=(n_repere%INTITS_PAGE==0 ? n_repere/INTITS_PAGE : (n_repere/INTITS_PAGE)+1);
min_v=(n_repere%INTITS_PAGE==0 ? INTITS_PAGE : n_repere%INTITS_PAGE);
ecran2(5, (mode==3 ? nivk : nivb), (* p_u));
fonction((mode==3 ? nivk : nivb), -1, (*p_v));
while(1)
switch(car=acquerir( -1, mode%2))
{
case _ESC:if((car=getchar())=='[')
    switch(car=getchar())
    {
        case 'D' : (* p_v) -= 1;
            if((* p_v)>=1) fonction((mode==3 ? nivk : nivb), (* p_v)+1, (* p_v));
            else
            {
                (* p_u) -= 1;
                if((* p_u)>=1)
                {
                    ecran2(5, (mode==3 ? nivk : nivb), (* p_u));
                    (* p_v)=INTITS_PAGE;
                    fonction((mode==3 ? nivk : nivb), -1, (* p_v));
                }
            }
        else
        {
            (* p_u) += 1;
            (* p_v) += 1;
        }
    }
}
}

```



```

        };
        break;
case 'C' : (* p_v) += 1;
        if ((* p_v) <= ((* p_u) == max_u ? min_v : INTITS_PAGE))
            fonction((mode == 3 ? nivk : nivb), (* p_v) - 1, (* p_v));
        else
        {
            (* p_u) += 1;
            if ((* p_u) <= max_u)
            {
                ecran2(5, (mode == 3 ? nivk : nivb), (* p_u));
                (* p_v) = 1;
                fonction((mode == 3 ? nivk : nivb), -1, (* p_v));
            }
            else
            {
                (* p_v) -= 1;
                (* p_u) -= 1;
            }
        }
        break;
default : break;
};
break;

case _ret : return;
default : break;
};
}

```

/*

```

*/
traiter_groupe()
{
    short i;
    short j;
    short niv;
    char car;
    char type;
    char option;

    p_liste=deb_liste;
    while(p_liste->suivant!=0)
    {
        p_liste=p_liste->suivant;
        free(p_liste);
    };
    deb_liste->suivant=0;

    adr_groupe      =i_groupe;
    pos              =4;
    modif[niv_deb-1]=0;

    ecran3(1," ", " ", " ");
    for(option=acquerir( -1,1); (option!='o')&&(option!='n'); option=acquerir( -1,1));

    _red_fle(BOE, _BYT, _WND_LGT, (i_boe=adr_deb-2));
    for(i= -1; i_boe<1_boe-2; )
        switch(type=get_chr(2))
        {
            case '*': if(niv==niv_deb) traiter_med('a');
                      if((niv>niv_deb)&&(option=='o')) traiter_med('a');
                      if((niv>niv_deb)&&(option=='n'))
                      {
                          while((car=get_chr(2))!=':');
                          while((car=get_chr(2))!=_CRT);
                      };
                      break;
            case '+':
            case '&': niv=(car=get_chr(2))- '0';

                      if((niv==niv_deb)&&(i>0))
                      {
                          _sve_chr('/');
                          return;
                      };
                      if(niv==niv_deb) i=traiter_intitule(niv);
                      if((niv>niv_deb)&&(option=='o')) traiter_intitule(niv);
                      if((niv>niv_deb)&&(option=='n'))
                      {
                          while((car=get_chr(2))!=':');
                          while((car=get_chr(2))!=_CRT);
                      };
                      break;
        };
    _sve_chr('/');
}

```

```

traiter_liste()

```

```

{
    char car;

```



```

{
char car;

adr_groupe=i_groupe;

_red_fle(BOE, _BYT, _WND_LGT, (i_boe=adr_deb));
while((car=get_chr(2))!='*');
traiter_med('k');
_sve_chr('/');
}

```

```

traiter_intitule(niv)
short niv;
{
short i;
short j;
short n;
char car;

symptome[j=0]='&';
if(niv+niv-niv_deb>9) symptome[++j]=(n=(niv+niv-niv_deb)/10)+'0';
symptome[++j]=(n=(niv+niv-niv_deb)%10)+'0';
for(i=0,j++;)
{
car=get_chr(2);
if((niv==niv_deb)&&(i>=0))
{
if(car==intit_fin[i]) i++;
else i=-1;
};
if(car==_CRT) car=SPACE;
symptome[j++]=car;
if(car==' ')
{
symptome[j++]=_CRT;
symptome[j]='\\0';
while((car=get_chr(2))!=_CRT);
break;
};
};
niveau(niv);
abreviation();
scission();
for(j=0; symptome[j]!='\\0'; j++) _sve_chr(symptome[j]);
return(i);
}

```

```

niveau(niv)
short niv;
{
char car;
short m;
short n;
short n1;
short n2;
short i;

```

```

short j;
char nbr[L_ECRAN];
char sym[L_SYMPT];

m=nivk+niv-niv_deb;
if(modif[niv-1])
{
    pos++;
    aff(m, symptome, 0, 1, (m>9 ? 2 : 1));
    for(ecran3(2, " ", " ", " ");)
    {
        for(i= -1; (car=acquerir(i, 1))!=_ret; )
            if((car>='0')&&(car<='9')) nbr[++i]=car;
        if(i== -1) n=m;
        else
        {
            if(i==0) n=nbr[i]-'0';
            else
            {
                n1=nbr[i]-'0';
                i--;
                n2=nbr[i]-'0';
                n=n2*10 + n1;
            }
        }

        if(n==m)
        {
            modif[niv]=0;
            break;
        }
        if((n>m)&&(n<=der_niv+1))
        {
            modif[niv]=1;
            break;
        }
    }

    for(i=0; symptome[i]!='\0'; i++) sym[i]=symptome[i];
    sym[i]='\0';
    symptome[j=0]=sym[0];
    if(n>9) symptome[++j]=(n2=n/10)+'0';
    symptome[++j]=(n2=n%10)+'0';
    for(i=(m>9 ? 2 : 1); sym[i]!='\0'; ) symptome[++j]=sym[++i];
    m=n;
    aff(m, symptome, 0, -1, -1);
}
else
{
    pos++;
    aff(m, symptome, 0, -1, -1);
    modif[niv]=0;
}
der_niv=m;
max_niv=(der_niv>max_niv ? der_niv : max_niv);
}

```

```

abbreviation()
{

```

```

char corr[512];
char abre[20];
char sym[1] SYMPT1.

```



```

char corr[512];
char abre[20];
char sym[L_SYMPT];
char car;
short j;
short k;
short l;

for(j=0; symptome[j]!='\0'; j++)
if(symptome[j]=='.')
{
    for(j--; (_UPR_CAS_CHR(symptome[j])) ||
              (_LWR_CAS_CHR(symptome[j])) ||
              (symptome[j]=='-')); j--);
    for(j++,l=0; symptome[j]!='.'; j++,l++) abre[l]=symptome[j];
    abre[l]='.';

    p_liste=deb_liste;
    for(k=0; abre[k]==p_liste->abre[k]; k++);
    if((abre[k]=='.')&&(p_liste->abre[k]!='\0'));
    else
    {
        aff(der_niv, symptome, 0, j-1, j);
        while(1)
        {
            if(p_liste->suiwant==0)
            {
                ecran3(4,corr,abre," ");
                p_liste->suiwant=(struct liste *)malloc(sizeof(struct liste));
                p_liste=p_liste->suiwant;
                for(k=0; abre[k]!='.'; k++) p_liste->abre[k]=abre[k];
                p_liste->abre[k]='\0';
                for(k=0; corr[k]!='\0'; k++) p_liste->corr[k]=corr[k];
                p_liste->corr[k]='\0';
                p_liste->suiwant=0;
                break;
            };
            p_liste=p_liste->suiwant;
            for(k=0; abre[k]==p_liste->abre[k]; k++);
            if((abre[k]=='.')&&(p_liste->abre[k]!='\0'))
            {
                for(ecran3(3,p_liste->corr,p_liste->abre," "),car=acquerir(-1, 1);
                    (car!='o')&&(car!='n');
                    car=acquerir(-1,1));
                if(car=='o') break;
                if(car=='n')
                {
                    ecran3(4,corr,abre," ");
                    for(k=0; corr[k]!='\0'; k++) p_liste->corr[k]=corr[k];
                    p_liste->corr[k]='\0';
                    break;
                };
            };
        };
        for(j=0; symptome[j]!='.'; j++) sym[j]=symptome[j];
        for(j--,k=j; (_UPR_CAS_CHR(sym[k])) ||
                    (_LWR_CAS_CHR(sym[k])) ||
                    (sym[k]=='-')); k--);
        for(k++,l=0; p_liste->corr[l]!='\0'; k++,l++) sym[k]=p_liste->corr[l];
        for(j+=2,l=k; symptome[j]!='\0'; j++,l++) sym[l]=symptome[j];
    }
}

```

```

sym[l]='\0';
for(l=0; sym[l]!='\0'; l++) symptome[l]=sym[l];
symptome[l]='\0';
j=k-1;
aff(der_niv, symptome, 0, -1, -1);
};
};

```

```

scission()

```

```

{
short i;
short j;
short k;
short l;
short m;
short i_rel;
char sym[L_SYMPT];
char car;

for(i=(der_niv>9 ? 3 : 2); symptome[i]==SPACE; i++);
for(j=i, i_rel=0; symptome[j]!='\0'; j++)
if((symptome[j]=='(') || (symptome[j]=='))')
if(j==i)
{
switch(symptome[j])
{
case '[':for(k=j; symptome[k+1]!='\0'; k++) symptome[k]=symptome[k+1];
symptome[k]='\0';
break;
case '(':for(k=j; symptome[k+1]!=')'; k++) symptome[k]=symptome[k+1];
for(; symptome[k+2]!='\0'; k++) symptome[k]=symptome[k+2];
symptome[k]='\0';
break;
};
aff(der_niv, symptome, i_rel, -1, -1);
j--;
}
else
{
aff(der_niv, symptome, i_rel, j, -1);
for(ecran3(5, " ", " ", " ", " "), car=acquerir(-1,1);
(car!='o') && (car!='n');
car=acquerir(-1,1));
switch(car)
{
case 'n':switch(symptome[j])
{
case '[':for(k=j; symptome[k+1]!='\0'; k++) symptome[k]=symptome[k+1];
symptome[k]='\0';
break;
case '(':for(k=j; symptome[k+1]!=')'; k++) symptome[k]=symptome[k+1];
for(; symptome[k+2]!='\0'; k++) symptome[k]=symptome[k+2];
symptome[k]='\0';
break;
};
aff(der_niv, symptome, i_rel, -1, -1);
j--;
}
}
}

```

```

break;

```

```

case 'n':modif(der_niv-niv,deh1=1;

```



```

        break;

case 'o': modif[der_niv-nivk+niv_deb]=1;
        for(k=0; k<j; k++) sym[k]=symptome[k];
        sym[k++]=':';
        sym[k++]='_CRT';
        aff(der_niv, sym, i_rel, -1, -1);
        sym[k++]='&';
        if(der_niv+1>9) sym[k++]=(m=(der_niv+1)/10)+'0';
        sym[k++]=(m=(der_niv+1)%10)+'0';
        der_niv += 1;
        max_niv =(der_niv>max_niv ? der_niv : max_niv);
        i_rel=(der_niv>9 ? k-3 : k-2);
        switch(symptome[j])
        {
        case '[':for(l=j; symptome[l+1]!='\0'; l++,k++) sym[k]=symptome[l+1];
                sym[k]='\0';
                break;
        case '(':for(l=j; symptome[l+1]!=')'; l++,k++) sym[k]=symptome[l+1];
                for(; symptome[l+2]!='\0'; l++,k++) sym[k]=symptome[l+2];
                sym[k]='\0';
                break;
        };
        for(k=0; sym[k]!='\0'; k++) symptome[k]=sym[k];
        symptome[k]='\0';
        pos++;
        aff(der_niv, symptome, i_rel, -1, -1);
        break;
};
};
}

```

```

_sve_chr(car)
char car;
{
    putc(car,fgroupe);
    i_groupe++;
}

```

```

traiter_med(cas)
char cas;
{
    char car;
    char med[L_MED];
    char cod[2];
    char deg;
    short upper;
    short lower;
    short i;

    switch(cas)
    {
    case 'a':_sve_chr('*');
            break;
    }
}

```

```

case 'k':break;
};
for(deg == '0', car=get_chr(2);;)
switch(car)
{
case SPACE:
case _CRT:
case ')':
case ',':car=get_chr(2);
break;
case '!':deg=correspondance('4');
car=get_chr(2);
break;
case '[':deg=correspondance('3');
car=get_chr(2);
break;
case '(' :deg=correspondance('1');
car=get_chr(2);
break;
case ':':switch(cas)
{
case 'a':_sve_chr(':');
_sve_chr(_CRT);
while((car=get_chr(2))!=_CRT);
return;
case 'k':return;
};
default :for(i=lower=upper=0;
((car!=SPACE)&&(car!=_CRT)&&(car!='')&&(car!=','))
&&(car!='!')&&(car!='[')&&(car!='(')&&(car!=':'));
i++,car=get_chr(2))
{
if(_LWR_CAS_CHR(car)) lower=1;
if(_UPR_CAS_CHR(car)) upper=1;
med[i]=(_UPR_CAS_CHR(car) ? car+32 : car);
};
med[i]='\0';

if(code(med,cod))
switch(cas)
{
case 'a':_sve_chr(cod[0]);
_sve_chr(cod[1]);
_sve_chr(deg=='0' ?
(((upper==1)&&(lower==0)) ?
correspondance('5') : correspondance('2')) : deg);
/*
_sve_chr('[');
_sve_chr('B');
_sve_chr(']');
*/
break;
case 'k':for(i=0; med[i]!='\0'; i++) _sve_chr(med[i]);
_sve_chr('\0');
_sve_chr(deg=='0' ?
(((upper==1)&&(lower==0)) ?
correspondance('5') : correspondance('2')) : deg);
/*
_sve_chr('[');
_sve_chr('B');
_sve_chr(']');
*/
}

```



```

        _sve_chr('1');
        /*
        break;
    };
    deg='0';
    break;
};
};

```

```

correspondance(deg)
char deg;
{
    switch(deg)
    {
        case '5':return('3');
        case '4':
        case '3':return('2');
        case '2':
        case '1':return('1');
    };
}

```

```

code(med,cod)
char med[];
char cod[];
{
    short i_buf;
    short i;
    char car;

    if(med[0]=='\0') return(0);

    p_rep1=deb_rep1->suivant;
    while(p_rep1->car!=med[0])
    {
        if(p_rep1->suivant==0)
        {
            ecran3(6," ", " ", med);
            return(code(med,cod));
        };
        p_rep1=p_rep1->suivant;
    };

    if(_BYT_LOC(MEDCO)!=p_rep1->loc)
        _red_fle(MEDCO, _BYT, LGR_MEDCO, p_rep1->loc);
    for(i_buf= -1; p_rep1->loc+i_buf<l_medco-2;)
    {
        if(medco[++i_buf]!=med[0]) break;
        for(i=0; medco[++i_buf]==med[++i];);
        if((medco[i_buf]=='*')&&(med[i]=='\0'))
        {
            for(i=0,i_buf++; medco[i_buf]!='*'; i++, i_buf++)
                med[i]=medco[i_buf];
            med[i]='\0';
            cod[0]=medco[++i_buf];
            cod[1]=medco[++i_buf];
        }
    }
}

```

```
        return(1);  
    };  
    while (medco[++i_buf] != _CRT);  
    };  
    ecran3(6, " ", " ", med);  
    return (code(med, cod));  
}
```

/*

*/
representer()


```

    */
    representer()
    {
        struct des * p_desprec;
        short i;

        determiner(nivk);
        p_des = deb_des;
        while(p_des->suiwant!=0)
        {
            p_desprec=p_des;
            p_des = p_des->suiwant;
            for(i=1; i<=NIV_MAX; i++)
                if(p_des->lock[i]== -1)
                {
                    i--;
                    break;
                };
            if(p_des->lock[i]>adr[0])
            {
                inserer(p_desprec,p_des);
                return;
            };
            if(p_des->lock[i]==adr[0])
            switch(type)
            {
                case 'N':
                    inserer(p_desprec,p_des);
                    return;

                case 'F':
                    if((p_des->typk=='N') ||
                       ((p_des->typk=='F') && (p_des->nivk>nivk)));
                    else
                    {
                        inserer(p_desprec,p_des);
                        return;
                    };
                    break;

                case 'E':
                    if((p_des->typk=='N') || (p_des->typk=='F'));
                    else
                    {
                        inserer(p_desprec,p_des);
                        return;
                    };
                    break;

                case '*':
                    if(p_des->typk=='*') return;
                    if((p_des->typk=='N') || (p_des->typk=='F') || (p_des->typk=='E'));
                    break;
            };
        };
        inserer(p_des,0);
    }

```

```

determiner(niv)
{
    short niv;
    {
        char car;
        long adr_deb;
        long adr_s_dic;
        short nbr_s_rub;
        short i;

        switch(type)
        {
            case '*':
            case 'E': for(i=1; i<=niv; i++)      adr[i]=nemok[i].adr;
                     for(i=niv+1; i<=NIV_MAX; i++) adr[i]= -1;
                     adr[0]=nemok[niv].adr;
                     return;
            case 'N': adr[0]=nemok[niv-1].adr+2;
                     break;
            case 'F': adr_deb=(niv<=3 ? -1 :
                     nemok[4].adr_dic-(nemok[4].lgr_dic-nemok[4].nbr_rub)
                     *_RCD_DEF(DN_KEN));

                     adr_s_dic=nemok[niv].adr_dic;
                     nbr_s_rub=nemok[niv].nbr_rub;
                     while(nbr_s_rub>0)
                     {
                         charger_dic(niv, (nbr_s_rub-1)/DIM, adr_s_dic, nbr_s_rub);
                         switch(niv)
                         {
                             case 1:
                             case 2:
                                 p_ken123=(struct ken123 *)
                                 (_BUF(D1_KEN+niv-1)+((nbr_s_rub-1)%DIM)*_RCD_DEF(D1_KEN+niv-1));
                                 adr_s_dic=p_ken123->adr_s_dic;
                                 nbr_s_rub=p_ken123->nbr_s_rub;
                                 break;

                                 case 3:
                                 p_ken123=(struct ken123 *)
                                 (_BUF(D3_KEN)+((nbr_s_rub-1)%DIM)*_RCD_DEF(D3_KEN));
                                 adr_deb=p_ken123->adr_s_dic*_RCD_DEF(DN_KEN);
                                 adr_s_dic=adr_deb
                                 +(p_ken123->lgr_s_dic-p_ken123->nbr_s_rub)*_RCD_DEF(DN_KEN);
                                 nbr_s_rub=p_ken123->nbr_s_rub;
                                 break;

                                 default:
                                 p_kenn=(struct kenn *)
                                 (_BUF(DN_KEN)+((nbr_s_rub-1)%DIM)*_RCD_DEF(DN_KEN));
                                 adr_s_dic=adr_deb+p_kenn->adr_s_dic*_RCD_DEF(DN_KEN);
                                 nbr_s_rub=p_kenn->lgr_s_dic;
                                 break;
                         }
                         niv++;
                     }
                     niv--;
                     adr[0]=(niv<=3 ? p_ken123->adr_rub+2 : p_kenn->adr_rub+2);
                };

        _red_fle(RUB, _BYT, _WND_LGT, adr[0]);
    }
}

```

```

car=_get_nxt_chr(RUB);
while(adr[0]<l_rub-2)
    ,

```



```

car=_get_nxt_chr(RUB);
while(adr[0]<1_rub-2)
{
    car=_get_nxt_chr(RUB);
    adr[0]++;
    if((car=='&')||(car=='+'))
    {
        niv=(car=_get_nxt_chr(RUB))-'0';
        for(i=1; i<niv; i++) adr[i]=nemok[i].adr;
        adr[niv]=adr[0];
        for(i=niv+1; i<=NIV_MAX; i++) adr[i]= -1;
        return;
    };
};
adr[1]=adr[0];
for(i=2; i<=NIV_MAX; i++) adr[i]= -1;
}

```

```

inserer(precedent,suivant)
struct des * precedent;
struct des * suivant;
{
    struct des * p;
    short i;

    precedent->suivant=(struct des *)malloc(sizeof(struct des));
    p=precedent->suivant;
    for(i=1; i<=NIV_MAX; i++) p->lock[i]=adr[i];
    p->typk    =type;
    p->nivk    =niv;
    p->acrg    =adr_groupe;
    p->suivant=suivant;
}

```

/*

```

*/
inserer_groupes()
{
    struct des * p_desprec;
    short i;
    short i_buf;
    char med[L_MED];
    char ref[L_REF];
    char deg;
    char cod[3];
    char buf[11];
    char car;

    f_groupe=open(nom_groupe,0);

    i_ken=0;

    p_des=deb_des;
    while(p_des->suiwant!=0)
    {
        p_desprec=p_des;
        p_des=p_des->suiwant;
        for(i=1; (i<=NIV_MAX)&&(p_des->lock[i]!=-1); i++)
            if(p_desprec->lock[i]!=p_des->lock[i]) aller_en(i,p_des->lock[i]+2);

        switch(p_des->typk)
        {
            case 'N':
            case 'F':
            case 'E':
                lseek(f_groupe, p_des->adrg, 0);
                read(f_groupe, buf, 1);
                while(buf[0]!='/')
                {
                    _sve_crt_chr(KENbis,buf[0]);
                    printf("%c",buf[0]);
                    read(f_groupe, buf, 1);
                };
                break;

            case '*':
                while((car=get_chr(3))!=':')
                {
                    _sve_crt_chr(KENbis,car);
                    printf("%c",car);
                };
                _sve_crt_chr(KENbis,':');
                printf(":");
                _sve_crt_chr(KENbis,(car=get_chr(3)));
                printf("%c",car);

                p_med=deb_med;
                while(p_med->suiwant!=0)
                {
                    p_med=p_med->suiwant;
                    free(p_med);
                };
                deb_med->suiwant=0;

                if((car=get_chr(3))== '*')

```

```

{
    car=get_chr(3);
    ...
}

```



```

{
car=get_chr(3);
while(car!=';')
{
cod[0]=car;
cod[1]=get_chr(3);
cod[2]='\0';
p_rep2=deb_rep2->suiwant;
while(p_rep2->car!=cod[0]) p_rep2=p_rep2->suiwant;
if(_BYT_LOC(MEDCO)!=p_rep2->loc) _red_fle(MEDCO, _BYT, LGR_MEDCO, p_rep2->loc);
for(i_buf=-1;;)
{
while(medco[++i_buf]!='*');
while(medco[++i_buf]!='*');
for(i=-1; medco[++i_buf]==cod[++i];);
if((medco[i_buf]==_CRT)&&(cod[i]=='\0'))
{
for(i_buf-=4; medco[i_buf]!='*'; i_buf--);
for(i_buf++,i=0; medco[i_buf]!='*'; i_buf++, i++)
med[i]=medco[i_buf];
med[i]='\0';
deg =get_chr(3);
car=get_chr(3);
if(car=='[')
{
for(i=0; car!=']'; i++, car=get_chr(3))
ref[i]=car;
ref[i]=']';
ref[++i]='\0';
car=get_chr(3);
}
else ref[0]='\0';
inserer_med(med,deg,ref);
break;
}
while(medco[++i_buf]!='_CRT');
};
};
car=get_chr(3); /*pour le _CRT */
}
else _red_fle(KEN, _BYT, _WND_LGT, (i_ken-=1));

lseek(f_groupe, p_des->adrg, 0);
read(f_groupe, buf, 1);
while(buf[0]!='/')
{
for(i=0; buf[i]!='\0'; i++,read(f_groupe, buf, 1))
med[i]=buf[i];
med[i]='\0';
read(f_groupe, buf, 1);
deg=buf[0];
read(f_groupe, buf, 1);
if(buf[0]=='[')
{
for(i=0; buf[i]!=']'; i++, read(f_groupe, buf, 1))
ref[i]=buf[i];
ref[i]=']';
ref[++i]='\0';
read(f_groupe, buf, 1);
}
}

```

```

        else    ref[0]='\\0';
        inserer_med(med, deg, ref);
    };

    _sve_crt_chr(KENbis, '*');
    printf("*");
    p_med=deb_med;
    while(p_med->suivant!=0)
    {
        p_med=p_med->suivant;
        p_rep3=deb_rep3->suivant;
        while(p_rep3->car!=p_med->med[0]) p_rep3=p_rep3->suivant;
        if(_BYT_LOC(MEDCO)!=p_rep3->loc)
            _red_fle(MEDCO, _BYT, LGR_MEDCO, p_rep3->loc);
        for(i_buf= -1;;)
        {
            for(i= -1; medco[++i_buf]==p_med->med[++i];);
            if((medco[i_buf]=='*')&&(p_med->med[i]=='\\0'))
            {
                _sve_crt_chr(KENbis,medco[++i_buf]);
                printf("%c",medco[i_buf]);
                _sve_crt_chr(KENbis,medco[++i_buf]);
                printf("%c",medco[i_buf]);
                _sve_crt_chr(KENbis,p_med->deg);
                printf("%c",p_med->deg);
                for(i=0; p_med->ref[i]!='\\0'; i++)
                {
                    _sve_crt_chr(KENbis,p_med->ref[i]);
                    printf("%c",p_med->ref[i]);
                };
                break;
            }
            while(medco[++i_buf]!='_CRT');
            while(medco[++i_buf]!='*');
        };
        _sve_crt_chr(KENbis, ':');
        printf(":");
        _sve_crt_chr(KENbis,_CRT);
        printf("\n");
        break;
    };

    /*
    printf(" _fin copie de groupe return\n");
    scanf("%c",&ret);
    */

    };

while(i_ken<=l_ken-2)
{
    _sve_crt_chr(KENbis,(car=get_chr(3)));
    printf("%c",car);
};
_sve_crt_chr(KENbis,_EOF_MRK);
close(f_groupe);
}

```



```

allier_en(n,loc)
short n;
long loc;
{
char objec[512];
char intit[512];
char caro;
char cari;
char type;
char car;
short o;
short i;
short niv;

if(lcc==1_rub)
{
while(i_ken<=l_ken-2)
{
car=get_chr(3);
printf("%c",car);
_sve_crt_chr(KENbis,car);
};
_sve_crt_chr(KENbis,_CRT);
printf("\n");
}
else
{
_red_fle(RUB, _BYT, _WND_LGT, loc);
for(o=0,car=_get_nxt_chr(RUB); car!=': '; o++,car=_get_nxt_chr(RUB) ) objec[o]=car;
objec[o]=': ';

while(i_ken<=l_ken-2)
switch(car=get_chr(3))
{
case '*':
_sve_crt_chr(KENbis,car);
printf("%c",car);
while((car=get_chr(3))!=': ')
{
_sve_crt_chr(KENbis,car);
printf("%c",car);
};
_sve_crt_chr(KENbis,': ');
printf(":");
break;

case '+':

case '&':
type =car;
niv =(car=get_chr(3))-'0';
if(niv==n)
for(o= -1, i= -1;;)
{
for(o++; (objec[o]==SPACE) || (objec[o]==_CRT); o++);
for(intit[++i]=get_chr(3); (intit[i]==SPACE) || (intit[i]==_CRT); intit[++i]=get_chr(3));
caro=(_UPR_CAS_CHR(objec[o]) ? objec[o]+32 : objec[o]);
cari=(_UPR_CAS_CHR(intit[i]) ? intit[i]+32 : intit[i]);

if((caro==':')&&(cari==':'))

```

```

{
    /*
    B_bell(1);
    B_put_screen();
    scanf("%c",&ret);
    */

    i_ken -= (i+3);
    _red_fle(KEN, _BYT, _WND_LGT, i_ken);
    return;
};

if (caro!=cari)
{
    _sve_crt_chr(KENbis,type);
    printf("%c",type);
    _sve_crt_chr(KENbis,(car=niv+'0'));
    printf("%c",car);
    if(intit[i]==':')
    {
        for(i=0; intit[i]!=': '; i++)
        {
            _sve_crt_chr(KENbis, intit[i]);
            printf("%c",intit[i]);
        };
        _sve_crt_chr(KENbis,': ');
        printf(": ");
    }

    else
    {
        intit[++i]='\0';
        for(i=0; intit[i]!='\0'; i++)
        {
            _sve_crt_chr(KENbis, intit[i]);
            printf("%c",intit[i]);
        };
        while((car=get_chr(3))!=': ')
        {
            _sve_crt_chr(KENbis,car);
            printf("%c",car);
        };
        _sve_crt_chr(KENbis,': ');
        printf(": ");
    };

    break;
};

else
{
    _sve_crt_chr(KENbis,type);
    printf("%c",type);
    _sve_crt_chr(KENbis,(car=niv+'0'));
    printf("%c",car);
    while((car=get_chr(3))!=': ')
    {
        _sve_crt_chr(KENbis,car);
        printf("%c",car);
    };
    _sve_crt_chr(KENbis,': ');
    printf(": ");

    break;
};

```



```

        break;
    };
    default:
        _sve_crt_chr(KENbis,car);
        printf("%c",car);
        break;
};
};
}

```

```

insérer_med(med,deg,ref)
char med[];
char deg;
char ref[];
{
    short i;

    p_med=deb_med;
    while(p_med->suivant!=0)
    {
        p_medprec=p_med;
        p_med=p_med->suivant;
        switch(comparer(med,p_med->med))
        {
            case '=':return;
            case '>':break;
            case '<':p_medprec->suivant=(struct med *)malloc(sizeof(struct med));
                    p_medprec=p_medprec->suivant;
                    for(i=0; med[i]!='\0'; i++) p_medprec->med[i]=med[i];
                    p_medprec->med[i] ='\0';
                    p_medprec->deg    =deg;
                    for(i=0; ref[i]!='\0'; i++) p_medprec->ref[i]=ref[i];
                    p_medprec->ref[i] ='\0';
                    p_medprec->suivant=p_med;
                    return;
        };
    };
    p_med->suivant=(struct med *)malloc(sizeof(struct med));
    p_med=p_med->suivant;
    for(i=0; med[i]!='\0'; i++) p_med->med[i]=med[i];
    p_med->med[i] ='\0';
    p_med->deg    =deg;
    for(i=0; ref[i]!='\0'; i++) p_med->ref[i]=ref[i];
    p_med->ref[i] ='\0';
    p_med->suivant=0;
}

```

```

comparer(med1,med2)
char med1[];
char med2[];
{
    short i1;
    short i2;

    for(i1=0, i2=0;;)

```

```

if (med1[i1]==med2[i2])
{
    if (med1[i1]=='\0') return('=');
    i1++;
    i2++;
}
else
{
    if ((med1[i1]=='.')||(med1[i1]=='\0')) return('<');
    if ((med2[i2]=='.')||(med2[i2]=='\0')) return('>');
    if ((med1[i1]!='-')&&(med2[i2]!='-'))
    {
        if (med1[i1]>med2[i2]) return('>');
        else return('<');
    };
    if (med1[i1]=='-') i1++;
    if (med2[i2]=='-') i2++;
};
}

```

/*

*/
 epilogue(cas)


```

*/
epilogue(cas)
{
    short cas;
    {
        short index;
        short i;

        switch(cas)
        {
            case 1:
                _cls_fle(BOE);
                _cls_fle(D1_BOE);
                _cls_fle(D2_BOE);
                _cls_fle(D3_BOE);
                _cls_fle(DN_BOE);
                _cls_fle(MEDCO);
                _cls_fle(KEN);
                _cls_fle(D1_KEN);
                _cls_fle(D2_KEN);
                _cls_fle(D3_KEN);
                _cls_fle(DN_KEN);
                _cls_fle(RUB);
                _cls_fle(KENbis);

                fclose(fgroupe);

                _opn_fle(INTER _IN_MODE _OUT);
                p_des=deb_des;
                while(p_des->suiwant!=0)
                {
                    p_des=p_des->suiwant;
                    index=_idx_inc(INTER, _RCD, _OUT);
                    for(i=1; i<=NIV_MAX; i++) inter[index].lock[i]=p_des->lock[i];
                    inter[index].typk=p_des->typk;
                    inter[index].nivk=p_des->nivk;
                    inter[index].adrg=p_des->adrg;
                };
                index=_idx_inc(INTER, _RCD, _OUT);
                inter[index].lock[0]=i_groupe;
                inter[index].lock[1]=max_niv;
                inter[index].nivk = -1;
                _idx_inc(INTER, _RCD, _EOF_MRK);
                _cls_fle(INTER);
                break;

            case 2:
                fclose(fgroupe);
                break;

            case 3:
                _cls_fle(BOE);
                _cls_fle(D1_BOE);
                _cls_fle(D2_BOE);
                _cls_fle(D3_BOE);
                _cls_fle(DN_BOE);
                _cls_fle(MEDCO);
                _cls_fle(KEN);
                _cls_fle(D1_KEN);
                _cls_fle(D2_KEN);
                _cls_fle(D3_KEN);

```

```
_cle_+le(DN_KEN);
_cle_fle(RUB);
_cle_fle(KENbis);
```

```
fgrupe=fopen(nom_groupe, "w");
fclose(fgrupe);
```

```

_oprn_fle(INTER _IN_MODE _OUT);
_index=_idx_inc(INTER, _RCD, _OUT);
inter[index].lock[0]= 0;
inter[index].lock[1]= 0;
inter[index].nivk    = -1;
_idx_inc(INTER, _RCD, _EOF_MRK);
_cle_fle(INTER);
break;
};
}

```

[illegible]

[illegible]

Request id: pr19-3199 Printer: pr19

Tue May 17 11:33:01 GMT 1988

```
/*
```

```
Directory : /usr2/demotte  
Intitule  : ecran1.c  
Compile   : cc corps1.c ecran1.c -les -o et3.exe  
Regroupe toutes les procedures dites d'ecran du programme charge de la  
mise a jour du fichier de la premiere version encodee du repertoire de  
KENT par insertion d'elements tires du fichier de la version encodee du  
repertoire de BOENNINGHAUSEN.
```

```
*/
```

```
#include "/usr/archimed/lib/iodef.c"  
#include "/usr/archimed/lib/iodec.c"  
#include "/usr2/demotte/def.c"
```

```
acquerir(i,mode)  
{  
    short i;  
    short mode;  
    {  
        char car;  
  
        B_clear(_LINE, _RGT, 20, 1, (mode ? 3+i : L_ECRAN+3+i));  
        B_put_screen();  
        B_locate(20, (mode ? 3+i : L_ECRAN+3+i));  
        B_put_screen();  
        putchar(car=getchar());  
        return(car);  
    }  
}
```

```
aff(niv, symptome, i0, fd, ff)  
{  
    short niv;  
    char symptome[];  
    short i0;  
    short fd;  
    short ff;  
    {  
        extern short pos;  
        extern short nivk;  
        short i;  
  
        pos=(pos==19 ? 5 : pos);  
        B_clear(_LINE, _RGT, pos, (pos==5 ? 16 : 1), 1);  
        B_clear(_LINE, _RGT, 21, 3, 1);  
        B_put_screen();  
        B_locate(pos, niv-nivk+2);  
        B_put_screen();  
        printf(">");  
        for(i=i0; symptome[i]!=':'; i++)  
        {  
            if(i==fd)  
            {  
                B_reverse();  
                B_put_screen();  
            };  
            printf("%c",symptome[i]);  
            if(i==ff)  
            {
```

```
                B_normal();  
                B_put_screen();  
            }
```



```

        B_normal();
        B_put_screen();
    };

    printf(":");
    B_normal();
    B_put_screen();
}

```

```

fonction(niv, i0, i1)
    short niv;
    short i0;
    short i1;
    {
        extern short mode;
        extern char symptome[];
        short i;
        short j;

        if(i0!= -1)
        {
            B_clear(_LINE, _RGT, 4+niv+i0, 1, (mode%2 ? 1 : L_ECRAN));
            B_put_screen();
            B_locate(4+niv+i0, (mode%2 ? niv+2 : L_ECRAN+niv+2));
            B_put_screen();
            for(i=1, j= -1; i<i0; i++) while(symptome[++j]!='*');
            printf(">");
            while(symptome[++j]!='*') printf("%c",symptome[j]);
        };
        B_clear(_LINE, _RGT, 4+niv+i1, 1, (mode%2 ? 1 : L_ECRAN));
        B_put_screen();
        B_locate(4+niv+i1, (mode%2 ? niv+2 : L_ECRAN+niv+2));
        B_reverse();
        B_put_screen();
        for(i=1, j= -1; i<i1; i++) while(symptome[++j]!='*');
        printf(">");
        while(symptome[++j]!='*') printf("%c", symptome[j]);
        B_normal();
        B_put_screen();
    }

```

```

ecran1(cas)
    short cas;
    {
        char * s;

        B_clear(_SCREEN, _ALL, 0, 0, 0);
        B_put_screen();
        s="Mise a jour du repertoire de Kent";
        B_locate(1,1);
        B_sprintf(s);
        s="
        B_locate(22,2);
        B_underscore();
        B_sprintf(s);
        B_normal();
    }

```

";

```

switch(cas)
{
case 1: s="options : i(nsertion) f(usion) a(rret)";
        break;
case 2: s="options : i(mmediate) r(eportee)";
        break;
};
B_locate(23,2);
B_sprintf(s);
B_put_screen();
}

```

```

ecran2(cas,niv,uple)

```

```

short cas;
short niv;
short uple;
{
char * s;
short j;

```

```

switch(cas)
{

```

```

case 1:

```

```

B_clear(_LINE, _RGT, 21, 3, 1);

```

```

B_put_screen();

```

```

s="

```

```

B_locate(21,2);

```

```

B_underscore();

```

```

B_sprintf(s);

```

```

B_normal();

```

```

s="mode niveau _options : < > _valeur :

```

```

B_locate(22,2);

```

```

B_reverse();

```

```

B_sprintf(s);

```

```

B_normal();

```

```

s="mode intitule _options : return /quitter

```

```

B_locate(23,2);

```

```

B_sprintf(s);

```

```

B_put_screen();

```

```

visionner_niv(niv);

```

```

break;

```

```

case 2:

```

```

B_clear(_SCREEN, _ALL, 0, 0, 0);

```

```

B_put_screen();

```

```

s="Mise a jour du repertoire de KENT";

```

```

B_locate(1,1);

```

```

B_sprintf(s);

```

```

s="Delimitation/Localisation dans BOENNINGHAUSEN ";

```

```

B_locate(3,2);

```

```

B_sprintf(s);

```

```

B_put_screen();

```

```

for(j=1; j<niv; j++) visionner_intit('D',j,0);

```

```

s="

```

```

B_locate(21,2);

```

```

B_underscore();

```

```

B_sprintf(s);

```

```

B_normal();

```

```

s="mode niveau _options : < > _valeur :

```

```

B_locate(22,2);

```

```

B_sprintf(s);

```



```

s="mode niveau      _options :    <    >                _valeur :
B_locate(22,2);
B_sprintf(s);
s="mode intitule  _options :    return    /quitter
B_locate(23,2);
B_reverse();
B_sprintf(s);
B_normal();
B_put_screen();
visionner_niv(niv);
break;

case 3:
B_clear(_LINE, _RGT, 4, 20-4, L_ECRAN);
B_clear(_LINE, _RGT, 20, 4, 1);
B_put_screen();
for(j=1; j<niv; j++) visionner_intit('D',j,0);
s="
B_locate(21,2);
B_underscore();
B_sprintf(s);
B_normal();
s="mode intitule  _options :    return    /quitter
B_locate(22,2);
B_sprintf(s);
B_put_screen();
break;

case 4:
B_clear(_SCREEN, _ALL, 0, 0, 0);
B_put_screen();
s="Mise a jour du repertoire de KENT";
B_locate(1,1);
B_sprintf(s);
s="Localisation dans KENT de l'insertion";
B_locate(3,2);
B_sprintf(s);
B_put_screen();
for(j=1; j<niv; j++) visionner_intit('D',j,0);
s="
B_locate(21,2);
B_underscore();
B_sprintf(s);
B_normal();
s="mode niveau      _options :    <    >                _valeur :
B_locate(22,2);
B_sprintf(s);
s="mode intitule  _options :    return    /quitter
B_locate(23,2);
B_reverse();
B_sprintf(s);
B_normal();
B_put_screen();
visionner_niv(niv);
break;

case 5:
visionner_intit('S',niv,uple);
B_clear(_LINE, _RGT, 21, 3, 1);
B_put_screen();

```

```

s="
B_locate(21,2);
B_underscore();
B_sprintf(s);
B_normal();
s="options : < > return";
B_locate(22,2);
B_sprintf(s);
B_put_screen();
break;
};
}

```

```

visionner_niv(niv)
short niv;
{
B_locate(22,62);
B_put_screen();
printf("%d",niv);
}

```

```

visionner_blanc(niv)
short niv;
{
extern short mode;

B_clear(_LINE, (mode/3 ? _RGT : _LFT), 4+niv, 20-(4+niv), (mode/3 ? 1 : L_ECRAN));
B_put_screen();
}

```

```

visionner_intit(cas,niv,uple)
char cas;
short niv;
short uple;
{
extern short mode;
extern struct nemo nemob[];
extern struct nemo nemok[];
extern struct boe123 * p_boe123;
extern struct boen * p_boen;
extern struct ken123 * p_ken123;
extern struct kenn * p_kenn;
extern struct ORDRE * deb_ORDRE;
extern struct ORDRE * p_ORDRE;
extern struct ordre * p_ordre;
extern char symptome[];
char car;
short i;
short j;
short k;
short l;
short m;

switch(cas)

```

```

{
case '0':
B_clear(_LINE, _RGT, 4+niv, 20-(4+niv), (mode/3 ? 1 : L_ECRAN));

```



```

{
case 'D':
B_clear(_LINE, _RGT, 4+niv, 20-(4+niv), (mode%2 ? 1 : L_ECRAN));
B_put_screen();
B_locate(4+niv, (mode%2 ? niv+1 : L_ECRAN+niv+1));
B_put_screen();
printf(">");
_red_fle((mode/3 ? RUB : BOE), _BYT, _WND_LGT,
(mode/3 ? nemok[niv].adr+2 : nemob[niv].adr));
for(i=niv+1; i<L_ECRAN; i++)
    switch(car=_get_nxt_chr(mode/3 ? RUB : BOE))
    {
    case '[' :break;
    case _CRT:
    case '?':return;
    default :if(_UPR_CAS_CHR(car)) car +=32;
              printf("%c",car);
              break;
    };
break;

case 'Z':
B_clear(_LINE, _RGT, 4+niv, 20-(4+niv), (mode%2 ? 1 : L_ECRAN));
B_put_screen();
B_bell(1);
B_locate(4+niv, (mode%2 ? niv+1 : L_ECRAN+niv+1));
B_put_screen();
printf(">non trouve");
break;

case 'L':
B_clear(_LINE, _RGT, 4+niv, 20-(4+niv), (mode%2 ? 1 : L_ECRAN));
B_put_screen();
B_locate(4+niv, (mode%2 ? niv+1 : L_ECRAN+niv+1));
B_put_screen();
printf(">a la suite de ");
charger_dic(niv,
(nemok[niv].nbr_rub-1)/DIM,
nemok[niv].adr_dic,
nemok[niv].nbr_rub);
if(niv<=3)
p_ken123=(struct ken123 *)(_BUF(D1_KEN+niv-1)+((nemok[niv].nbr_rub-1)%DIM)*_RCD_DEF(D1_KEN+niv-1));
else
p_kenn=(struct kenn *)(_BUF(DN_KEN)+((nemok[niv].nbr_rub-1)%DIM)*_RCD_DEF(DN_KEN));
_red_fle(RUB, _BYT, _WND_LGT, (niv<=3 ? p_ken123->adr_rub+2 : p_kenn->adr_rub+2));
for(i=niv+1; i<L_ECRAN; i++)
    switch(car=_get_nxt_chr(RUB))
    {
    case _CRT:
    case '?':return;
    default :if(_UPR_CAS_CHR(car)) car+=32;
              printf("%c",car);
              break;
    };
break;

case 'N':
B_clear(_LINE, _RGT, 4+niv+1, 20-(4+niv+1), (mode%2 ? 1 : L_ECRAN));
B_put_screen();
B_locate(4+niv+1, (mode%2 ? niv+2 : L_ECRAN+niv+2));

```

```

B_put_screen();
printf(">derive de ");
_red_fle(RUB, _BYT, _WND_LGT, nemok[niv].adr+2);
for(i=niv+2; i<L_ECRAN; i++)
    switch(car=_get_nxt_chr(RUB))
    {
        case _CRT:
        case '?': return;
        default :if(_UPR_CAS_CHR(car)) car +=32;
                    printf("%c",car);
                    break;
    };
break;

case 'S':
B_clear(_LINE, _RGT, 4+niv, 20-(4+niv), (mode%2 ? 1 : L_ECRAN));
B_put_screen();
for(p_ORDRE=deb_ORDRE, j=0; j<uple; p_ORDRE=p_ORDRE->suiwant, j++);
p_ordre=p_ORDRE->p_ordre;

for(m= -1, l=1, j= -1; l<=INTITS_PAGE; l++)
{
    if((p_ordre->n_ordre/DIM)!=j)
    {
        j=p_ordre->n_ordre/DIM;
        charger_dic(niv,
            j,
            (mode/3 ? nemok[niv].adr_dic : nemob[niv].adr_dic),
            (mode/3 ? nemok[niv].nbr_rub : nemob[niv].nbr_rub));
    };
    if(niv<=3)
    {
        if(mode/3)
            p_ken123=(struct ken123 *)
                (_BUF(D1_KEN+niv-1)+(p_ordre->n_ordre%DIM)*_RCD_DEF(D1_KEN+niv-1));
        else
            p_boe123=(struct boe123 *)
                (_BUF(D1_BOE+niv-1)+(p_ordre->n_ordre%DIM)*_RCD_DEF(D1_BOE+niv-1));
        _red_fle((mode/3 ? RUB : BOE), _BYT, _WND_LGT,
            (mode/3 ? p_ken123->adr_rub+2 : p_boe123->adr_boe));
    }
    else
    {
        if(mode/3)
            p_kenn=(struct kenn *)
                (_BUF(DN_KEN)+(p_ordre->n_ordre%DIM)*_RCD_DEF(DN_KEN));
        else
            p_boen=(struct boen *)
                (_BUF(DN_BOE)+(p_ordre->n_ordre%DIM)*_RCD_DEF(DN_BOE));
        _red_fle((mode/3 ? RUB : BOE), _BYT, _WND_LGT,
            (mode/3 ? p_kenn->adr_rub+2 : p_boen->adr_boe));
    };
    B_locate(4+niv+1, (mode%2 ? niv+2 : L_ECRAN+niv+2));
    B_put_screen();
    printf(">");
    for(k=niv+1; (k>0)&&(k<L_ECRAN-1); k++)
        switch(car=_get_nxt_chr(mode/3 ? RUB : BOE))
        {
            case '[' :
            case _CRT: break;

```

```

        case '?': k= -1;
                    break;
        default :if(_UPR_CAS_CHR(car)) car+=32;

```



```

        case ':': :k= -1;
                    break;
        default :if(_UPR_CAS_CHR(car)) car+=32;
                    symptome[++m]=car;
                    printf("%c",car);
                    break;
    };
    symptome[++m]='*';

    if(p_ordre->suiwant==0) break;
    p_ordre=p_ordre->suiwant;
};
break;
};
}

```

```

ecran3(cas,corr,abre,med)
    short cas;
    char corr[];
    char abre[];
    char med[];
    {
        char * s;
        char car;
        short i;

        switch(cas)
        {
            case 1:
                B_clear(_SCREEN, _ALL, 0, 0, 0);
                B_put_screen();
                s="Mise a jour du repertoire de KENT";
                B_locate(1,1);
                B_sprintf(s);
                s="Traitement de l'ensemble a inserer";
                B_locate(3,2);
                B_sprintf(s);
                s="? prise en consideration des rubriques derivees";
                B_locate(5,2);
                B_sprintf(s);
                s="
                B_locate(22,2);
                B_underscore();
                B_sprintf(s);
                B_normal();
                s="options : o(ui) n(on)";
                B_locate(23,2);
                B_sprintf(s);
                B_put_screen();
                break;

            case 2:
                s="
                B_locate(21,2);
                B_underscore();
                B_sprintf(s);
                B_normal();
                s="? niveau : return";

```

```

B_locate(22,2);
B_sprintf(s);
B_put_screen();
break;

case 3:
B_clear(_LINE, _RGT, 21, 3,1);
B_put_screen();
s="
B_locate(21,2);
B_underscore();
B_sprintf(s);
B_normal();
B_locate(22,2);
B_put_screen();
printf("? ");
for(i=0; abre[i]!='\0'; i++) printf("%c",abre[i]);
printf(". est abreviation de ");
for(i=0; corr[i]!='\0'; i++) printf("%c",corr[i]);
printf(". : o(ui) n(on)");
break;

```

```

case 4:
B_clear(_LINE, _RGT, 21, 3, 1);
B_put_screen();
s="
B_locate(21,2);
B_underscore();
B_sprintf(s);
B_normal();
s="? correspondance de l'abreviation ";
B_locate(22,2);
B_sprintf(s);
B_put_screen();
for(i=0; abre[i]!='.'; i++) printf("%c",abre[i]);
B_put_screen();
for(i= -1;;)

```

```

{
B_clear(_LINE, _RGT, 23, 1, 5+i);
B_put_screen();
B_locate(23, 5+i);
B_put_screen();
putchar(car=getchar());
switch(car)
{
case _ret:printf("\n");
corr[++i]='\0';
return;
case _DEL:if(i>=0) i--;
break;
default :corr[++i]=car;
break;
};
};

```

```
break;
```

```

case 5:
s="
B_locate(21,2);
B_underscore();

```

```

B_sprintf(s);
B_normal();

```



```

B_sprintf(s);
B_normal();
s="? scission : o(ui) n(on)";
B_locate(22,2);
B_sprintf(s);
B_put_screen();
break;

case 6:
B_clear(_LINE, _RGT, 21, 3, 1);
B_put_screen();
s="
B_locate(21,2);
B_underscore();
B_sprintf(s);
B_normal();
s="? correction du remede ";
B_locate(22,2);
B_sprintf(s);
B_put_screen();
for(i=0; med[i]!='\0'; i++) printf("%c",med[i]);
printf(" :");
for(i=-1;;)
{
B_clear(_LINE, _RGT, 23, 1, 5+i);
B_put_screen();
B_locate(23, 5+i);
B_put_screen();
putchar(car=getchar());
switch(car)
{
case _ret:printf("\n");
med[++i]='\0';
return;
case _DEL:if(i>=0) i--;
break;
default :med[++i]=(_UPR_CAS_CHR(car) ? car+32 : car);
break;
};
};
};

```



=SB10101/1988/04/2